(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 September 2003 (18.09.2003)

PCT

(10) International Publication Number
WO 03/077588 A1

(51) International Patent Classification^7: H04Q 7/38, H04L 12/56

(21) International Application Number: PCT/SE02/00468

(22) International Filing Date: 13 March 2002 (13.03.2002)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **TELEFONAKTIEBOLAGET L M ERICSSON** [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MALOMSOKY, Szabolcs** [HU/HU]; Szechenyi-tér 31, 3. Em. 9, H-2000 Szentendre (HU). **NÁDAS, Szilveszter** [HU/HU]; Ifjúsági 1tp. 15. 3/20, H-5400 Mezötúr (HU). **RÁCZ, Sándor** [HU/HU]; X. Kovago u. 6, H-1108 Budapest (HU).

(74) Agents: **HEDBERG, Åke** et al.; Aros Patent AB, P.O. Box 1544, S-751 45 Uppsala (SE).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

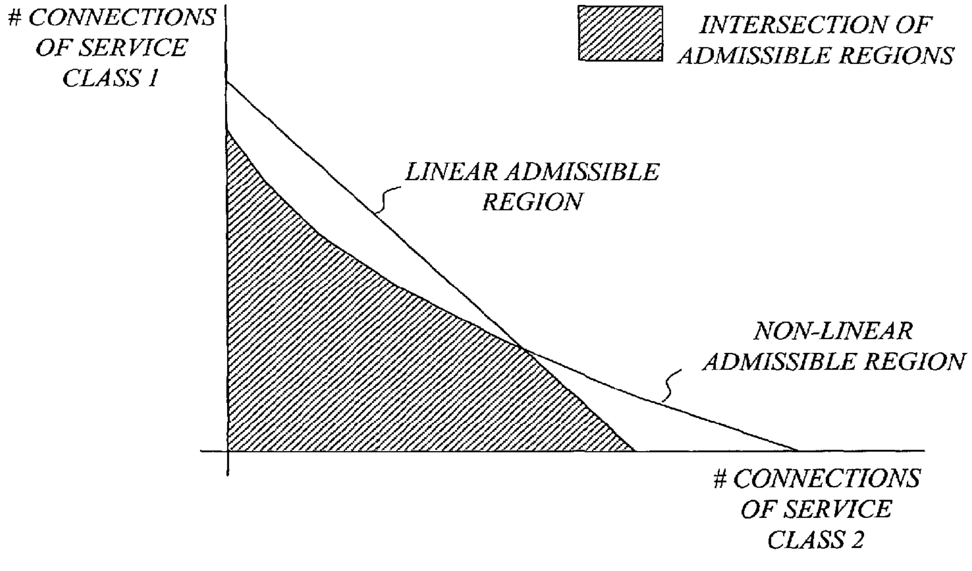
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *of inventorship (Rule 4.17(iv)) for US only*
— *of inventorship (Rule 4.17(iv)) for US only*
— *of inventorship (Rule 4.17(iv)) for US only*

Published:
— *with international search report*

[Continued on next page]

(54) **Title:** CONNECTION ADMISSION CONTROL IN PACKET-ORIENTED, MULTI-SERVICE NETWORKS



(57) **Abstract:** The present invention is generally based on the recognition that the true admissible regions for a multi-service traffic mix can be well approximated by a construction of a non-linerar admissible region and one or more linear admissible regions. This makes it possible to accurately control admission of a new connection onto a transport link by checking whether the multi-service traffic mix defined by previously admitted connections together with the new connection is contained within an intersection on a non-linear admissible region and at least one linear admissible region, and admitting the connection if the traffic mix is contained within the intersecton of regions.

— *with amended claims*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

CONNECTION ADMISSION CONTROL IN PACKET-ORIENTED, MULTI-SERVICE NETWORKS

TECHNICAL FIELD OF THE INVENTION

The present invention generally relates to connection admission control and more particularly to connection admission control in packet-oriented, multi-service networks with relatively strict delay and loss requirements.

BACKGROUND OF THE INVENTION

Connection admission control (CAC) is generally a question of controlling the number of connections using a given set of resources in a communication network, thereby ensuring that admitted connections have access to the resources that are required to fulfil their Quality of Service (QoS) requirements. On the link level, CAC serves to restrict the number of connections simultaneously present on a transport link in the network. This means that new connections may be rejected in order to protect connections that are already admitted for transport over the link.

The issue of connection admission control in packet-oriented networks with limited transmission resources and relatively strict delay and loss requirements, such as higher generation radio access networks, is generally quite complex. With the introduction of multi-service networks, such as Universal Mobile Telecommunications System (UMTS) and similar communication networks, it becomes even more difficult to find an efficient CAC strategy that works in the multi-service environment and at the same time fulfils practical requirements such as limited computational complexity and high accuracy.

The practical requirements on the CAC algorithm mainly imply that the CAC decisions need to be taken fast, because hundreds or thousands of connections may arrive to a network node each second, and that the CAC algorithm has to make relatively accurate estimates of the resource requirements so that the CAC decisions are not too conservative, nor too optimistic.

In the prior art, connection admission control is fairly simple, based on the concept of effective bandwidths. This generally means that each individual connection is assigned a bandwidth value that represents the "effective" resource usage of the connection during its lifetime. When a new connection arrives to the node, the effective bandwidth of the connection is estimated based on factors such as the traffic characteristics and the QoS requirements. Subsequently, the CAC algorithm checks whether the sum of effective bandwidths of the admitted connections and the new connection exceeds the link capacity or not. The algorithm is so simple that the CAC decisions can be taken on-line. This approach thus satisfies the requirement on limited computational complexity.

Unfortunately, CAC based on effective bandwidths is generally not capable of ensuring that the QoS requirements in a multi-service traffic environment are actually fulfilled since the associated set or region of admissible traffic mixes with a single linear boundary is not a sufficiently accurate estimate of the true admissible region. The linear admissible region obtained from the effective-bandwidth algorithm is highly dependent on how well the assigned effective bandwidths represent the actual resource usage of the connections, and even a slight misestimation of the required resources may result in QoS degradations (underestimation) or a significant waste of valuable resources (overestimation).

RELATED ART

A comparison of different connection admission control algorithms in ATM/AAL2 based third generation mobile access networks can be found in reference [1].

Reference [2] relates to a connection admission control strategy for ATM core switches and describes an effective-bandwidth algorithm for constant-bit-rate (CBR) connections as well as a connection admission control approach for variable-bit-rate (VBR) traffic. In the latter case, when statistically multiplexible VBR traffic (S-VBR) and non-statistically multiplexible VBR traffic (NS-VBR) are intermixed, the admissible region boundary is approximated by two piecewise linear segments corresponding to respective cell-loss-limited regions.

Reference [3] consider the problem of bounding the loss rates of the aggregation of ON-OFF sources in a bufferless model.

SUMMARY OF THE INVENTION

The present invention overcomes these and other drawbacks of the prior art arrangements.

It is a general object of the present invention to provide an efficient connection admission control strategy for a multi-service traffic environment.

It is particularly important that the connection admission control is highly accurate, thus providing optimized resource utilization, while still ensuring that the QoS requirements of the entire multi-service traffic mix are met. It is thus an object of the invention to estimate the true admissible region as accurately as possible.

It is also an object of the invention to provide a computationally efficient connection admission control algorithm, thus allowing on-line decisions on the acceptance of new connections.

These and other objects are met by the invention as defined by the accompanying patent claims.

The present invention is generally based on the recognition that the true admissible region for a multi-service traffic mix can be well approximated by a construction of a non-linear admissible region and one or more linear admissible regions. This makes it possible to accurately control admission of a new connection onto a transport link by checking whether the multi-service traffic mix defined by previously admitted connections together with the new connection is contained within an intersection of the non-linear admissible region and the linear admissible region(s), and admitting the connection only if the traffic mix is contained within the intersection of regions.

By properly identifying the non-linear admissible region and the linear admissible region or regions according to the QoS requirements of the mixed services and the traffic characteristics, accurate decisions on the admittance of new connections can be taken to optimize the resource utilization while ensuring the QoS requirements.

In general, the admissible regions in the construction of the "true" admissible region are related to respective QoS requirements such as packet delay and overload (packet loss).

In radio access networks such as the Universal Terrestrial Radio Access Network (UTRAN), and especially on those links connecting base stations and radio network controllers, the use of delay-limited linear admissible regions in combination with an overload-limited non-linear admissible region has turned out to be particularly beneficial.

For improved performance and flexibility, each service class in the multi-service traffic mix is preferably associated with a class-specific delay-limited admissible region. Each class-specific delay-limited admissible region is normally defined as a linear admissible region that contains a set of traffic mixes that fulfil a given class-specific packet delay requirement.

Advantageously, the overload-limited non-linear admissible region contains the set of traffic mixes for which the probability of temporarily overloading the queuing system associated with the transport link is smaller than a given target value.

It has been validated that the linear approximation of the delay-limited admissible regions is very accurate. The linearity of these admissible regions means that the evaluation of whether a given traffic mix is contained within each of the delay-limited regions can be performed in a computationally efficient manner, because efficient single-class approximations can be extended to multiple service classes.

The non-linearity of the overload-limited admissible region generally implies that the probability of overload must be evaluated individually for each traffic mix. In many cases, this results in far too heavy calculations for on-line evaluation. However, by exploiting the so-called statistical gain within the different classes and not (or only partially) between classes, a much more computationally efficient algorithm is obtained.

In certain situations, it is not necessary to use both the linear and non-linear admissible regions. In fact, if the delay requirements are loose or the link capacities are large enough, it may be sufficient to check whether the traffic mix is contained within the non-linear admissible region. In other circumstances, it may be sufficient to use a construction of multiple linear admissible regions. In effect, the intersection of multiple linear admissible regions generally defines a non-linear region, which has a piecewise linear boundary.

The invention offers the following advantages:

- High accuracy, leading to optimized resource utilization and maintained QoS for the entire multi-service traffic mix; and
- Computational efficiency, allowing on-line decisions on the acceptance of new connections.

Other advantages offered by the present invention will be appreciated upon reading of the below description of the embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention, together with further objects and advantages thereof, will be best understood by reference to the following description taken together with the accompanying drawings, in which:

Fig. 1 illustrates the set of admissible traffic mixes of voice connections and 64 Kbps packet switched connections calculated using a conventional CAC algorithm, compared to the exact set of admissible mixes;

Fig. 2 is a schematic diagram illustrating an approximation of the true admissible region as a construction of a generally non-linear admissible region and one or more linear admissible regions according to a preferred embodiment of the invention;

Fig. 3 is a schematic diagram illustrating the basic architecture of a UMTS network;

Fig. 4 illustrates a typical probability density function of packet delay for a single service class;

Fig. 5 is a schematic diagram of an ATM/AAL2 based protocol stack of a UTRAN network;

Fig. 6 illustrates a network node fed by two ON-OFF connections;

Fig. 7 is a schematic diagram illustrating an example of an admissible region constructed as an intersection of two linear delay-limited regions and a non-linear overload-limited region;

Fig. 8 is a schematic flow chart of a method for connection admission control according to a preferred embodiment of the invention;

Fig. 9 is a schematic block diagram of pertinent parts of a network node in which a CAC algorithm according to the invention may be implemented;

Fig. 10A illustrates the admissible region in a first exemplary traffic environment with three service classes;

Fig. 10B illustrates a simulation of the delay violation probability for mixes on the surface of the admissible region of Fig. 10A;

Fig. 11A illustrates the admissible region obtained by the CAC of a preferred embodiment of the invention in a second exemplary traffic environment; and

Fig. 11B illustrates a simulation of the delay violation probability for mixes on the surface of the admissible region of Fig. 11A.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Throughout the drawings, the same reference characters will be used for corresponding or similar elements.

In general, the task of the connection admission control (CAC) function in a network node is to decide whether a new connection that arrives to the node can be accepted for transport over a link of a given capacity such that the Quality of Service (QoS) requirements of the new connection and the already admitted connections are not violated. In order to ensure that the amount of resources is sufficient to serve the traffic demands, and at the same time ensure that expensive resources are not wasted, reliable CAC methods are needed.

For a better understanding of the invention it may be useful to begin by describing the basic concept of conventional CAC based on effective bandwidths in more detail. For example, the CAC algorithm currently in use in the transport and control platform Cello from Ericsson assigns an effective bandwidth to each connection, and simply estimates the resource usage of all connections as the sum of their effective bandwidths. In multi-service networks, connections can normally be grouped into service classes based on their traffic descriptors, and it is thus possible to assign an effective bandwidth to each service class. When a new connection arrives to the Cello node, the effective bandwidth of the connection is calculated by means of a simple exponential formula. Subsequently, the following inequality is checked:

$$\sum_{i=1}^{N} EB_i \le C, \qquad (1)$$

where N is the number of connections in the traffic mix defined by the admitted connections and the new connection, the EB_i values are the effective bandwidths of the connections and C is the link capacity. In this way, it can be determined whether the sum of the effective bandwidths of the traffic mix under consideration exceeds the link capacity or not. If the estimated resource usage of the traffic mix does not exceed the link capacity, the new connection is admitted for transport on the associated link, otherwise the connection request is rejected.

However, experiments reveal that conventional CAC algorithms based on effective bandwidths are generally not capable of ensuring that the QoS requirements in a multi-service traffic environment are actually fulfilled since the linear admissible region thus obtained is not a sufficiently accurate estimate of the true admissible region, as schematically illustrated in Fig. 1.

Fig. 1 illustrates the set of admissible traffic mixes of voice connections and 64 Kbps packet switched connections calculated using a conventional CAC algorithm, compared to the exact set of admissible mixes. The admissible region estimated in accordance with the conventional CAC algorithm has a single linear boundary, whereas the true (exact) admissible region is generally non-linear. The discrepancy between the estimated admissible region and the true admissible region leads to unreliable CAC decisions, with either too many or too few connections being admitted. Although the admissible regions illustrated in Fig. 1 may seem to be relatively close to each other, a slight underestimation of the required resources may result in serious QoS degradations since the delay may increase very rapidly at high loads.

In contrast to the prior art, the invention preferably estimates the true admissible region by a construction of a generally non-linear admissible region and one or more linear admissible regions, as schematically illustrated for two service classes in Fig. 2. Mathematical evidence in combination with extensive simulations and experiments indicate that such a construction of non-linear and linear admissible regions is indeed a very good approximation of the true admissible region. This basic recognition makes it possible to accurately control admission of a new connection by checking whether a multi-service traffic mix defined by previously admitted connections together with the new connection is contained within an intersection of a generally non-linear admissible region and one or more linear admissible regions, and admitting the new connection only if the traffic mix is within the intersection of admissible regions.

An efficient CAC algorithm can be devised by properly identifying the non-linear admissible region and the linear admissible region or regions according to the QoS requirements of the mixed services and the traffic characteristics of the network under consideration. In this way, the resource utilization can be optimized while ensuring the QoS requirements.

In the following, the invention will mainly be described with respect to radio access networks such as the Universal Terrestrial Radio Access Network (UTRAN) in third generation mobile communication systems or other similar future communication systems.

With the introduction of third generation mobile systems, such as the Universal Mobile Telecommunications System (UMTS), both equipment vendors and network operators face new challenges. In contrast to second generation systems, a packet-switched, multi-service network capable of fulfilling the specific requirements of the new radio interface technology called Wideband Code Division Multiple Access (WCDMA) is required.

Fig. 3 is a schematic diagram illustrating the basic architecture of a UMTS network. The UMTS network 100 basically includes a core network 110, a Universal Terrestrial Radio Access Network (UTRAN) 120 and user equipment (UE) 130. The core network 110 is the backbone of UMTS connecting the access network 120 to external networks 200 such as the Public Switched Telephone Network (PSTN) and the Internet. The UTRAN network 120 handles all tasks related to radio access, and therefore UTRAN nodes are responsible for radio resource management, handover control, and so forth. The UTRAN network 120 is based on Radio Network Controllers (RNCs) 122 and base stations (Node Bs) 124. The user equipment 130 such as a mobile is connected to the Node B base stations 124 over the radio interface (Uu). A user terminal 130 can communicate with several Node Bs 124 at the same time during soft handover (which is an essential interference reduction technique in WCDMA systems). A Node B base station 124 is connected to an RNC 122 over the so-called Iub interface, whereas RNCs 122 are connected to each other over the so-called Iur interface. The RNCs 122 are connected to the core network 110 over the Iu interface.

Switching and multiplexing technologies used for the first releases of UTRAN for UMTS are based on Asynchronous Transfer Mode (ATM) in combination with the

ATM Adaptation Layer type 2 (AAL2). Future releases will be deployed using also Internet Protocol (IP) technologies.

On transport links of UTRAN radio access networks, and especially on those links connecting base stations and radio network controllers, resource allocation is complex because the Quality of Service (QoS) requirements are quite strict and the amount of transmission resources is relatively low.

UMTS is a multi-service network, in which the various service classes generally have different QoS requirements. Packet delay is usually the most important performance measurement in the network, and for each service class, the delay budget for the whole system (end-to-end) determines the maximum acceptable delay in the UTRAN transport network. For example, the end-to-end maximum delay requirement for voice traffic is around 180 ms, and the maximum delay requirement in the UTRAN transport network is around 5-6 ms. The delay requirements for other services are not very different from that of voice. In case best effort services such as those provided in IP based networks, the delay requirements are relatively strict because the use of soft handover sets a practical limit on the packet delays. In addition, studies focusing on TCP performance show that the application-level throughput is significantly degraded if the delay on the Iub interface is larger than a few milliseconds.

The delay requirement is typically defined in a probabilistic manner:

$$\Pr\left(D_i > \widetilde{D}_i\right) < \widetilde{\varepsilon}_i, \qquad (2)$$

where D_i is the delay of a packet of class-i, $\widetilde{D}_i$ is the maximum (target) delay for class-i, $\Pr\left(D_i > \widetilde{D}_i\right)$ is the probability of packet delay criteria violation and $\widetilde{\varepsilon}_i$ is a small target probability. A maximum delay of 5 ms and a target probability of 1% thus translates into the requirement that the probability of a delay larger than 5 ms should be smaller than 1%. Fig. 4 illustrates a typical probability density function of packet delay for a single service class, with the delay on the x-axis. The probability that the packet delay is larger than a given delay, say 5 ms, is represented by the tail area of the density function from the given delay.

The invention will now be described with respect to a particular example of a UTRAN network based on ATM/AAL2.

Fig. 5 is a schematic diagram of an ATM/AAL2 based protocol stack of a UTRAN network. The retransmission mechanism of the Radio Link Control (RLC) protocol ensures reliable transmission of loss-sensitive traffic over the radio interface. The Medium Access Control (MAC) protocol forms radio frames and schedules these periodically according to the timing requirements of WCDMA. This frame scheduling period is called *TTI* (Transmission Time Interval), and its length is typically a multiple of 10 ms. Bit rates of radio connections, so-called Radio Access Bearers (RABs) currently take typical values between 8 Kbps and 384 Kbps, keeping in mind that higher bit rates are feasible. MAC frame sizes and *TTI* lengths are typically RAB-specific. If the user equipment 130 has simultaneous RABs to two or more Node Bs 124 (during soft handover), the radio frames scheduled in downlink have to be sent out from every Node B 124 to the user equipment 130 at the same time (t_{out}). Therefore, nodes must generally be synchronized. For the same reason, it has to be ensured that each frame arrives to the Node Bs before t_{out}. This determines a delay requirement on the UTRAN transport network. On the Iub interface, the start positions of frames intended for different user terminals should not coincide in time in order to decrease the probability of packet congestions in the transmission network queues. Moreover, on the Uu interface, control patterns such as pilot bits should not be transmitted at the same time for all user terminals to avoid peaks in the interference. Therefore, phases of the periodic frame flows of different connections are randomly distributed over the Transmission Time Interval, as noted in reference [4].

The role of the UTRAN network is basically to transport MAC frames from the RNCs 122 to the Node Bs 124 (in the Iub downlink direction) and to transport MAC frames from Node Bs 124 to RNCs 122 (in the Iub uplink direction). In the UTRAN transport network, MAC frames are encapsulated into Iub frames. The Iub framing overhead contains information used in Node Bs 124 to encode the frame into the appropriate radio frame format and to send it out on the radio interface at the right time (t_{out}). Iub frames are segmented and packed into AAL2 CPS (Common Part Sublayer) packets, which are multiplexed into ATM cells. By AAL2 multiplexing, several AAL2 packets from different connections can be carried within an ATM cell. In an ATM network, cells are transported along a predefined path using the VPI/VCI (Virtual Path and Virtual Circuit Identifier) fields in the ATM header. The CID (Connection Identifier) field in the AAL2 header identifies a specific AAL2 connection within an ATM VC. the so-called CU timer (T_{CU}) determines how long the multiplexer should wait for arriving AAL2 packets if an ATM cell is partly filled. Therefore, multiplexing efficiency also depends on the value of T_{CU}, as realized in reference [5].

For admission control, we focus on situations when the transport links are highly loaded. In this region, the effect of T_{CU} on delay performance is expected to be negligible. It is also assumed that the packing density does not depend on the Iub frame size. These assumptions are supported by the analysis provided in reference [6]. In UTRAN, a new AAL2 connection is set up for each new RAB. In general, the AAL2 CAC allocates resources for the new AAL2 connections in the transport network, and makes CAC decisions based on traffic descriptors and QoS parameters.

At this stage, it will be useful to introduce an adequate queuing model for the UTRAN transport network, with specific emphasis on the Iub interface.

An exemplary queuing model for the UTRAN network

As discussed earlier, the arrival pattern of Iub frames is determined by the MAC scheduler, which schedules MAC frames periodically according to the timing requirements of WCDMA. In other words, the traffic is shaped by the MAC scheduler such that the lowest time-scale behavior is periodic irrespectively of the type of application. UTRAN traffic can thus be modeled as a superposition of periodic traffic sources. However, the carried traffic is generally not seen as a continuous periodic packet flow, but rather the user/application level traffic model is reflected in the UTRAN transport network such that the carried traffic is modeled by a series of active and inactive intervals. These intervals will be referred to as ON (active) and OFF (inactive) periods. Fig. 6 illustrates a network node fed by two ON-OFF connections. The network node comprises a packet server 140 with an associated output link capacity C and a queue 150 of length B fed by two periodic ON-OFF connections with different packet inter-arrival times (*TTI*) and packet sizes. In an ON period, MAC frames are sent in each period, while in an OFF period, packets are not sent at all. For example, in case of voice traffic the characteristics of the ON and OFF periods are determined by the interaction of the speech process (the speaker behavior) and the voice activity detector in the voice coder. The ON-OFF behavior, which originates from the user behavior, can be taken into account by an “activity factor”. The activity factor is the ratio of the ON intervals, defined as the average length of ON periods divided by the sum of the average lengths of ON and OFF periods. An activity factor can not be used to characterize a single connection. It can however be used to characterize all connections belonging to a certain service class in the system. In other words, the activity factor is a statistical measure describing the user/application behavior. If the activity factor is less than 1, then there is a potential for statistical multiplexing gain. For example, empirical measurements show that the activity factor for voice could be set to around 0.7.

It is desirable to establish a model, which allows us to derive the probability of packet delay requirement violation under the consideration of delays due to:

- the ON-OFF behavior, which results in temporary system overloads; and

- the periodic packet emission during the ON (active) states, where the emission phases of different connections are uniformly distributed over the *TTI*s associated with the connections, which can result in packet congestions.

Assume that we have a system with the following input parameters:

K	number of service classes
C	link capacity
N_i	number of AAL2 connections (RABs) of class-i
α_i	activity factor of class-i
n_i	number of actually active connections of class-i
b_i	packet size (MAC frame with transport overhead) of class-i
TTI_i	packet inter-arrival time of class-i
$\tilde{D}_i$	maximum delay (target delay) of class-i
$\tilde{\varepsilon}_i$	target probability of delay requirement violation for class-i

Input parameters (i=1, ..., K)

The total delay of a packet from service class-i includes two parts: $D_i = W_i + S_i$, where S_i is the service time and W_i is the waiting time in the queue. The service time is the time taken by the packet server to service a packet and send it out of the buffer. For example, if the bit rate of the output link is 100 Kbps (kilobits per second) then the service time for a packet that is 100 bits long will be 1 ms. The delay of lost packets is considered to be infinite.

The workload (the amount of unfinished work) plays an important role in a multi-class queue analysis because in contrast with the per-class waiting time, workload is a global measure and allows us to calculate the per-class waiting times in case of a First-In-First-Out (FIFO) service discipline. In a FIFO queue the waiting time is approximated by the workload. The workload and accordingly the waiting time strongly depend on the lengths of the ON and OFF periods. If the system is in an overload situation in which the input rate R of active connections exceeds the link capacity C, the workload has an increasing component. When the overload situation ends, there will be a decreasing component.

In general, delays on a time-scale longer than the transmission time interval (*TTI*) have to be studied by examining the workload accumulated over several *TTI* periods. The accumulated workload is burst-length dependent and generally describes the component of the workload due to the ON-OFF behavior. The development of the accumulated workload depends on the random nature of the traffic sources, i.e. the distribution of the ON and OFF periods, the dependency among the traffic sources and so forth.

In UTRAN, the delay requirements are relatively strict, typically shorter than or of the same order as the *TTI*, and therefore we are generally interested in the short time-scale behavior of the workload. Besides, the signaled traffic descriptors directly available from the network do not include any characterization of burstiness thus gives no possibility to properly characterize the long time-scale behavior of the workload. It may be possible to measure the burst lengths. However, this is generally of little interest in the evaluation of the probability of delay requirement violation in UTRAN.

Before describing the actual model construction it will be useful to examine the impact of buffer size on the workload. Systems with large buffers can absorb the packets generated during overload situations, resulting in large waiting times without losses. On the other hand, a moderate size buffer ($\sim TTI_{max}$), where $TTI_{max} = \max\ [TTI_i]$, is filled up quickly in an overload situation, and then packets suffer from at most a waiting time corresponding to the queue length or they get lost. The delay requirements in UTRAN are relatively strict ($D \sim TTI$), which means that the waiting time reaches the predefined delay requirement very fast in an overload situation, and therefore the queue can not smooth out an overload situation efficiently even if considering infinite buffer lengths. We thus consider a system that can buffer packets at least the packet delay requirement but not significantly longer. It is furthermore assumed that in an overload situation, the delay of all packets is always larger than the delay requirement. Applying these assumptions we establish a combined model. Considering packets in the system, two types of delay requirement violation events can be observed:

- some packets are lost due to buffer overflow; and
- some packets are only exceeding their respective delay requirement.

The following relation expresses this decomposition:

$$\varepsilon_i = \frac{number\ of\ lost\ packets + number\ of\ delayed\ packets}{number\ of\ packets}, \qquad (3)$$

and we define two measures as:

$$\varepsilon_i^{lost} = \frac{number\ of\ lost\ packets}{number\ of\ packets}, \qquad (4)$$

$$\varepsilon_i^{delayed} = \frac{number\ of\ delayed\ packets}{number\ of\ packets}. \qquad (5)$$

Simulations and experiments show that the model assumptions are perfectly valid on the time-scale corresponding to the transmission time interval (*TTI*). In this region the delays hardly depend on the length of the ON periods, and delay violations are dominated by the periodic packet emission.

Now, we are in the position to determine the main relation between the input parameters and a relevant CAC decision. The stationary probability $\Pi(\underline{n})$ of a particular number of active connections $\underline{n} = (n_1, n_2, \ldots, n_K)$ is calculated using a multi-dimensional binomial distribution as follows:

$$\Pi(\underline{n}) = \prod_{i=1}^{K} \Pi_i(n_i) = \prod_{i=1} \binom{N_i}{n_i} \alpha_i^{n_i} (1-\alpha_i)^{N_i - n_i} . \qquad (6)$$

From the model assumptions, it follows that packet loss probability can be well approximated by overload probability. The probability ε_i^{lost} that a packet of class-i arrives at an overload situation ($R > C$) and gets lost, can thus be calculated as:

$$\varepsilon_i^{lost} = \frac{\sum_{R(\underline{n})>C} n_i \Pi(\underline{n})}{\sum_{\forall \underline{n}} n_i \Pi(\underline{n})}, \qquad (7)$$

where $R(\underline{n})$ represents the input rate of the active connections in a given state $\underline{n}$.

In case of a normal situation ($R \leq C$) the waiting time is dominated by the periodic packet emission, because the transient period (due to the change of the number of active sources when $R \leq C$) is approximately TTI_{max} long and hardly influences the waiting time. Therefore the probability of a packet arriving later than its delay criteria but not getting lost can be calculated as:

$$\begin{aligned} \varepsilon_i^{delayed} &= \frac{\sum_{R(\underline{n})\leq C} n_i \Pi(\underline{n}) \cdot \Pr(D_i > \widetilde{D}_i | \underline{N} = \underline{n})}{\sum_{\forall \underline{n}} n_i \Pi(\underline{n})} = \\ &= \sum_{R(\underline{n})\leq C} \Pi(\underline{n})' \cdot \Pr(D_i > \widetilde{D}_i | \underline{N} = \underline{n}). \end{aligned} \qquad (8)$$

Finally, the probability of delay criteria violation is the sum of the two probabilities as follows:

$$\varepsilon_i = \varepsilon_i^{lost} + \varepsilon_i^{delayed}. \qquad (9)$$

An exemplary CAC algorithm for the UTRAN network

When a new connection arrives, the CAC algorithm normally needs to check:

- the delay violation due to packet loss (overload)
- the delay violation due to delayed packets (delay)

In accordance with a preferred embodiment of the invention, the true admissible region is approximated by an intersection of K regions (also referred to as hyper-planes) with linear borders and a region with a generally non-linear border. The K regions with linear borders are delay-limited and referred to as delay-limited linear admissible regions. The i-th delay-limited linear region preferably contains the mixes where the delay requirement of the i-th traffic class is fulfilled. The region with a non-linear border is overload-limited and referred to as an overload-limited non-linear admissible region. The overload-limited non-linear region preferably contains the mixes for which the probability of temporarily overloading the queuing system is smaller than a given target value. If the activity factor of each class is 1, the overload-limited border becomes linear. In this case, it is the border of the mixes that do not overload the system.

For example, assume that we have two service classes with different delay requirements. We are interested in the region where traffic mixes containing connections from both services can be accepted, and the admissible region is defined as the intersection of delay-limited and overload-limited regions, as illustrated in

Fig. 7. One of the service classes has stricter delay requirements than the other class, and therefore only the strictest delay-limited region is considered. This delay-limited region contains the set of traffic mixes that fulfil a given packet delay requirement.

If the activity factors are equal to 1 for both service classes, the overload-limited region becomes linear and contains the mixes that do not overload the buffer. In practice, however, not all activity factors are equal to 1. This means that the overload-limited region generally will be non-linear, containing traffic mixes that can overload the queuing system only with a small probability. If the activity factor one or more of the service classes is smaller than 1 (which means the service is bursty on a time scale that is larger than its TTI), the overload-limited region generally becomes concave as schematically illustrated in Fig. 7.

The linear approximation of the delay-limited admissible regions is very accurate and means that the evaluation of whether a given traffic mix is contained within each of the delay-limited regions can be performed in a computationally efficient manner.

The non-linearity of the overload-limited admissible region generally implies that the probability of overload must be evaluated individually for each traffic mix.

Checking the delay violation due to packet loss – the overload-limited region

Although it is possible to use equation (7) to check the delay violation due to packet loss, the evaluation of equation (7) may be too demanding for on-line use (depending on the available processing resources). By exploiting the so-called statistical gain only within different classes and not between classes, a much more computationally efficient algorithm is obtained. Therefore, the following preferred strategy of the invention is proposed. Keeping in mind that loss probability and overload probability are more or less interchangeable according to the model assumptions, an upper bound for the target loss probability $\tilde{\varepsilon}_i^{lost}$ is derived:

$$1-\tilde{\varepsilon}_i^{lost} = \frac{\sum_{R(\underline{n})\le C} n_i \prod_{k=1}^{K} \Pi_k(n_k)}{\sum_{\forall \underline{n}} n_i \Pi(\underline{n})}, \tag{10}$$

$$1-\tilde{\varepsilon}_i^{lost} \ge \frac{\sum_{n_1=0}^{A_1} \sum_{n_2=0}^{A_2} \cdots \sum_{n_K=0}^{A_K} n_i\, \Pi_1(n_1)\Pi_2(n_2)\ldots\Pi_K(n_K)}{\sum_{\forall \underline{n}} n_i \Pi(\underline{n})}, \tag{11}$$

$$1-\tilde{\varepsilon}_i^{lost} \ge \frac{\sum_{n_1=0}^{A_1} \Pi_1(n_1)\cdot \sum_{n_2=0}^{A_2} \Pi_2(n_2)\ldots \sum_{n_i=0}^{A_i} \Pi_i(n_i)\ldots \sum_{n_K=0}^{A_K} \Pi_K(n_K)}{\sum_{n_i=0}^{N_i} n_i\, \Pi_i(n_i)}, \tag{12}$$

$$1-\tilde{\varepsilon}_i^{lost} \ge \frac{\sum_{n_i=0}^{A_i} n_i \prod_i(n_i)}{N_i \alpha_i} \cdot \prod_{l\ne i} \left(\sum_{n_l=0}^{A_l} \prod_l(n_l) \right), \tag{13}$$

where we introduce A_i as a per-class limit on the number of simultaneously active connections in ON state. Preferably, the per-class limit A_i is defined as the number of connections from class-i such that the probability that more than A_i connections from class-i are active at the same time is small. The idea is to "distribute" the probability $1-\tilde{\varepsilon}_i^{lost}$ over the terms on the right hand side of the last inequality and to find minimal values of A_i such that the following sets of inequalities:

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \ge \frac{\sum_{n_i=0}^{A_i} n_i \prod_i(n_i)}{N_i \alpha_i}, \tag{14}$$

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \ge \sum_{n_l=0}^{A_l} \prod_l(n_l) \qquad l=1,2,\ldots,K_a, l\ne i, \tag{15}$$

are fulfilled, where K_a is the number of classes with activity factor $\alpha_i < 1$ and $\tilde{\varepsilon}_i^{lost}$ is the target probability assigned to class-i. For traffic classes with $\alpha = 1$, $A_i = N_i$. This construction makes it possible to compute A_i values independently of each other, thus significantly reducing the computational complexity of equation (7). With enough memory for the CAC algorithm, one may wish to obtain the A_i values from tables calculated off-line, as will be described in more detail later on.

Using the per-class limits (A_1, A_2, ..., A_K), the necessary condition for accepting a given traffic mix (N_1, N_2, ..., N_K) is defined as:

$$\sum_{i=1}^{K} A_i \rho_i \leq C, \qquad (16)$$

where ρ_i is the average load generated by one active traffic source from class-i and defined as $\rho_i = b_i / TTI_i$. The evaluation of the inequalities (16) generally corresponds to the evaluation of whether the traffic mix is contained within the non-linear overload-limited region.

The statistical gains between classes can at least partially be taken into account by extending the basic algorithm in the following manner. It is clear that by using equations (14) and (15), only the statistical gain of multiplexing sources from the same class will be exploited. Keeping the property that A_i values can be obtained independently from each other, but taking into account partially statistical gains from multiplexing different classes one may proceed as follows:

1. Find A_i^* for all i using eq.(14) with:

$$N_i^* = N_i + \sum_{\alpha_k \leq \alpha_i, k \neq i} \min\left(1, \frac{\rho_k}{\rho_i}\right) N_k; \quad k = 1, ..., K, \qquad (17)$$

and calculate the statistical multiplexing gain for class-i as:

$$MG_i = \frac{(N_i^* - A_i^*)}{N_i^*}. \tag{18}$$

2. Repeat the following procedure until the MG_i values are no longer increasing:

- Consider the classes with $\alpha_k > \alpha_i$ and $\rho_k < \rho_i$, $\forall\ i, k$. If $MG_k > MG_i$, then let $\alpha_i' := \alpha_k$ and calculate MG_i' as described in step 1. If $MG_i' > MG_i$, then let $MG_i := MG_i'$. (19-A)

- Consider the classes with$\alpha_k > \alpha_i$ and $\rho_k \geq \rho_i$, $\forall\ i, k$. If $MG_k > MG$, then let $MG_i := MG_k$. (19-B)

3. Finally, the A_i values are found as:

$$A_i = N_i(1 - MG_i). \tag{20}$$

Checking the delay violation due to delayed packets – the delay-limited regions

There is no exact formula for evaluating the delay violation due to delayed packets, as recognized in reference [7]. We define the admissible region for checking the delay violation due to delayed (but not lost) packets by K hyper-planes. The i-th hyper-plane defines the region where the probability of a packet arriving later than its delay criteria but not getting lost is equal to or smaller than a given target probability, $\varepsilon_i^{delayed} \leq \tilde{\varepsilon}_i^{delayed}$. This means that each class-specific packet delay requirement requires that the probability of the class-specific packet delay being larger than a given class-specific maximum delay is smaller than a given target value. We propose a method for hyper-plane construction that uses only a single-class calculation for evaluating the corresponding delay violation. For this purpose, the following two measures are introduced:

- *TN_{ij}* is the maximum number of connections from class-i assuming that a single packet from class-j would fulfil the packet delay requirements of class-j. We preferably approximate by *TN_{ij}* the maximum number of connections from class-i if one additional connection from class-j is present in the system; and

- *TE_{ij}* is a service class equivalent measure representing how many new connections that can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j. For example, consider two service classes I and J. Imagine that there are 10 class-I connections and 20 class-J connections in the system. In this configuration, assume that 1% of the class-J packets are delayed more than 5 ms, and that this means that the delay requirement of class-J connections is just met. Take out one class-I connection from the system. In this case, the 1% percentile delay of class-J connections decreases below 5 ms. Next, add new class-J connections to the system until the 1% percentile delay of class-J connections reaches 5 ms again. For example, if the resulting number of class-J connections is 24, then $TE_{IJ} = 4$.

The analysis of a constant service-time queue fed by n_i homogeneous periodic (active) connections is mainly based on the results in reference [8], and the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \tilde{D}_j | n_i$ connections are active) is calculated in the following way:

$$\Pr(D_j^{(i)} > \tilde{D}_j | n_i \text{ connections are active}) = \\ = \sum_{x'<l\leq n_i} \binom{n_i}{l} \left(\frac{l-x'}{TTI'}\right)^{l} \left(1-\frac{l-x'}{TTI'}\right)^{n_i-l} \cdot \frac{TTI'-n_i+x'}{TTI'-l+x'}, \qquad (21)$$

where $D_j^{(i)}$ denotes the delay of a packet from class-j assuming that the delay of the associated queue comes from only class-i connections, $\tilde{D}_j$ is the target delay criteria of packets from class-j, and the following additional measures are introduced:

$$x' = (\tilde{D}_j - \frac{b_j}{C}) / TU, \tag{22}$$

$$TTI' = TTI_i / TU, \tag{23}$$

$$TU = \frac{b_i}{C}. \tag{24}$$

The proposed formula for determining the *TN* matrix is:

$$TN_{ij} = \max\left\{ N_i \,\middle|\, \sum_{n_i=0}^{N_i} \Pi(n_i)\ \Pr(D_j^{(i)} > \tilde{D}_j | n_i \text{ connections are active}) \le \tilde{\varepsilon}_j^{delayed} \right\}, \tag{25}$$

where the stability criteria is not needed, because it is included in the inequalities (16).

The *TE* matrix can be determined from the *TN* matrix as follows:

$$TE_{ij} = TN_{jj}/\ TN_{ij}. \tag{26}$$

Using traffic class equivalents, the necessary condition of accepting a given traffic mix $(N_1, N_2, \ldots, N_K)$ is defined as:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \le TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K, \tag{27}$$

If we approximate by TN_{ij} the maximum number of connections from class-i if one additional connection from class-j is present in the system, as proposed above, the constant should be set to 1. Using a fixed constant equal to 1 has proved to be important in the case of priority scheduling (as discussed later on), but can be used for FIFO scheduling as well. In a more conservative approach, the constant is set to zero, assuming that we approximate by TN_{ij} the maximum number of connections from class-i if no additional connection from class-j is present in the system.

The evaluation of the inequalities (27) corresponds to the evaluation of whether the traffic mix is contained within each of the linear delay-limited regions.

Alternatively, the probability of packet delay criteria violation can be approximated by the following expression for the class-specific complementary distribution function $Q_i(x)$ of the workload in a FIFO queue, using a Brownian bridge approximation [9, 10]:

$$Q_i(x) = \exp\left\{-\frac{2Cx}{TTI_i\, n_i\, \rho_i^2}\left(\frac{Cx}{TTI_i} + C - n_i\, \rho_i\right)\right\}, \tag{28}$$

$$x = \widetilde{D}_j - \frac{b_j}{C}. \tag{29}$$

Consequently, instead of using equations (21)-(24), the probability of packet delay criteria violation is now determined as:

$$\Pr(D_j^{(i)} > \widetilde{D}_j \mid n_i \text{ connections are active}) = \exp\left\{-\frac{2Cx}{TTI_i\, n_i\, \rho_i^2}\left(\frac{Cx}{TTI_i} + C - n_i\, \rho_i\right)\right\}, \tag{30}$$

$$x = \widetilde{D}_j - \frac{b_j}{C}. \tag{31}$$

Compared to equations (21)-(24), equations (30) and (31) do not involve a lot of summations and therefore the probabilities of packet delay criteria violation can be calculated much faster. Once the probabilities of packet delay criteria violation has been calculated according to equations (30) and (31), the TN and TE matrices can be determined according to equations (25) and (26) and the inequalities (27) can be checked.

In yet another alternative embodiment of the invention, the following even faster approximation for determining the *TN* matrix is used:

$$TN_{ij} = \left\lceil \frac{C + \dfrac{C x \alpha_i}{TTI_i}}{\alpha_i \rho_i - \dfrac{\alpha_i \rho_i^2 \, TTI_i \ln(\tilde{\varepsilon}_j^{delayed})}{2x}} \right\rceil, \qquad (32)$$

$$x = \tilde{D}_j - \frac{b_j}{C}. \qquad (33)$$

The TN matrix is thus determined according to equations (32) and (33), and the TE matrix is determined as usual according to equation (26). Once the TN and TE matrices are determined, the inequalities (27) may be checked.

Naturally, by using the above fast approximations, the accuracy will be slightly reduced. The trade-off between computational complexity and accuracy has to be carefully evaluated by the system designer for each particular application.

The introduced queuing model for ATM/AAL2 and the associated CAC calculations can be more or less directly applied also in IP based UTRAN networks provided that the IP networks are CAC-enabled. The above model and the related methodology are applicable irrespectively of the transport technology used by the network.

Flow chart

For a better overview of the CAC algorithm according to a preferred embodiment of the invention, reference is now made to the basic flow chart of Fig. 8. When a new connection arrives to a network node implementing the CAC algorithm according to the above preferred embodiment of the invention, it is determined (step S1) whether or not the connection belongs to a new service class based on signaled traffic descriptors.

It can be noted that since TN_{ij} and TE_{ij} depend only on the service classes present in the system, but not on the actual number of connections, these matrices are generally not necessary to calculate on-line. The TN and TE matrices are only updated when a new service class is added. Consequently, if the connection belongs to a new service class (Y) not yet included in the TN and TE matrices, the relevant traffic descriptors and QoS requirements of this service class are added (step S2) to the general information database used by the CAC algorithm and the TN and TE matrices are updated (step S3). This update can usually be performed rapidly without having to re-compute the whole matrices. However, if the normal TE and TN update proves to be too slow for on-line use, a fast update (step S4) of these matrices is required in order to be able to make a fast decision on the acceptance of the connection. Such a fast update may for example be based on peak bandwidths, or by using equations (32) and (33) above. On the other hand, if the connection belongs to a service class already present in the system (N), then the TN and TE matrices are read (step S5) and the delay violation due to delayed packets is checked (step S6) according to formula (27).

The A_i values depend on N_i, α_i, the target loss probabilities $\tilde{\varepsilon}_i^{lost}$ and the number K_a of traffic classes with activity factors $\alpha_i < 1$. The A_i values can be calculated on-line (step S7), or if on-line calculation turns out to be too slow, tables with pre-calculated values can be constructed in advance (step S8) and subsequently accessed during on-line evaluation. At least some A_i values can thus be pre-calculated for a range of different values of N_i and/or α_i ($\tilde{\varepsilon}_i^{lost}$), and stored in memory tables, as illustrated below.

α=0.1

N	$A(\sqrt{1-\tilde{\varepsilon}_i^{lost}})$	$A(\sqrt[2]{1-\tilde{\varepsilon}_i^{lost}})$	...	$A(\sqrt[K_\alpha]{1-\tilde{\varepsilon}_i^{lost}})$
1 2 . . . N^{max}				

α=0.2

N	$A(\sqrt{1-\tilde{\varepsilon}_i^{lost}})$	$A(\sqrt[2]{1-\tilde{\varepsilon}_i^{lost}})$	...	$A(\sqrt[K_\alpha]{1-\tilde{\varepsilon}_i^{lost}})$
1 2 . . . N^{max}				

.
.
.

α=0.9

N	$A(\sqrt{1-\tilde{\varepsilon}_i^{lost}})$	$A(\sqrt[2]{1-\tilde{\varepsilon}_i^{lost}})$	...	$A(\sqrt[K_\alpha]{1-\tilde{\varepsilon}_i^{lost}})$
1 2 . . . N^{max}				

A_i tables for different values of N and α.

This means that for a number of α values (e.g. 0.1, 0.2, ..., 0.9), tables with A_i values are stored, and for a certain $[\alpha_i, N_i]$ pair, an A_i value can be read from the corresponding table for a certain target probability. If the memory consumption is too high, only a certain window $[1, 2, \ldots, N_{max}]$ should be stored. This is normally sufficient, since the number of connections from a certain service class only changes in steps of one (+1: one connection arrives to the system, -1: one connection leaves the system). This means that the necessary window location and size can be tracked easily, and the tables can be updated whenever necessary. Other solutions using for example interpolation techniques can also be applied. Once the A_i values are obtained, the delay violation due to packet loss is checked (step S9) according to formula (16).

Finally, the results of the two delay violation checks are combined (step S10) to produce a final CAC decision. In this particular embodiment of the invention, a given traffic mix is accepted only if both sets of inequalities (16) and (27) are fulfilled.

If the update and the decision making parts are carefully separated, the decisions can surely be made on-line even for large systems with many traffic classes. However, the update speed needs to be checked carefully, because it is possible that also the updates can be performed on-line.

Example of implementation of an admission controller

Connection admission control is normally exercised over output link resources of network nodes, accepting or rejecting connections in accordance with a CAC algorithm. The CAC algorithm, may for example be implemented as hardware, software, firmware or any suitable combination thereof. In ATM networks, for example, traffic descriptors are signaled to the network nodes, and the nodes make decisions as to whether connections can be admitted based on the signaled information. In general, taking the network perspective, a connection is only admitted if it is accepted by all nodes taking part in the end-to-end transmission of that connection.

Fig. 9 is a schematic block diagram of pertinent parts of a network node in which a CAC algorithm according to the invention may be implemented. The network node 300 such as an RNC or a Node B is generally associated with a number of input links and output links. The node 300 preferably comprises a control unit 310, a switch fabric 315, a number of output buffers 320 and corresponding output servers 330. The control unit 310 is preferably built as a more or less embedded computer system with a processor 312 and associated memory system 314. The processor 312 may for example be a microprocessor or a digital signal processor. The CAC algorithm is preferably implemented as software executed by the control unit 310. The software may be written in almost any type of computer language, such as C, C++ or even specialized proprietary languages. In effect, the CAC algorithm is mapped into a software program, which when executed by the processor 312 produces CAC decisions in response to given QoS requirements and traffic information maintained in a special traffic information database 316 in the memory system 314. The traffic information is received from network traffic descriptors using for example classical signaling exchange mechanisms. Special look-up tables 318 for holding information used by the CAC algorithm may also be provided in the memory system 314. For each new connection set-up, the control unit 310 retrieves relevant information from the traffic information database and/or directly from the inter-node signaling, performs the necessary calculations and table look-ups and finally makes a CAC decision. The CAC decision is forwarded to the relevant protocol layers in the protocol stack to allow the decision to be effectuated. If a connection is rejected, a notification that the ATM/AAL2 or corresponding layer can not service the requested connection is sent up to the relevant protocol layers. If a connection is accepted, the connection is established and the corresponding packets are forwarded via the switch fabric 315 to the relevant buffer 320 for subsequent service by the corresponding output server 330. Internal traffic control messages from the control unit 310 generally controls packet switching and scheduling, and more particularly e.g. to which queue, if an admission decision is taken in a multi-queue system, the connection is directed. An example of a software implementation for execution by the control unit 310 is given in Appendix A.

Performance of the proposed CAC

The performance of the invention has been evaluated in simulations. In a first simulation, the following input parameters were considered: C=1920 Kbps, TTI_1=20ms, b_1=336 bits, α_1=0.65, TTI_2=10ms, b_2=1512 bits, α_2=0.85, TTI_3=40ms,

b_3=2688 bits, α_3=1, $\tilde{D}_1$=5ms, $\tilde{D}_2$=8ms, $\tilde{D}_3$=20ms, and $\tilde{\varepsilon}$=0.001. The admissible region obtained using the invention is illustrated in Fig. 10A. It can be seen that the borders of the admissible region are really hyper-planes. The surface covering the admissible region corresponds to those traffic mixes that just fulfil the QoS requirements, and for which a single extra connection would lead to a delay (or loss) violation.

The system with the traffic mixes on the surface was simulated to check if these mixes really fulfil the delay requirement. Fig. 10B illustrates the delay violation probability for the mixes on the surface of the admissible region of Fig. 10A. Identifiers of the different traffic mixes on the surface of the admissible region are given on the x-axis, and the probability of delay criteria violation is given on the y-axis. By way of example, traffic mix (N_1, N_2, N_3) = (3, 5, 6) has ID 45. For the purpose of examining the performance of the CAC, it is not important to tell which mix is associated with which ID. It can though be appreciated from Fig. 10B that the 0.001 limit is kept nicely for most of the mixes on the surface of the admissible region, implying that the CAC really works quite well.

The example of Figs. 11A-B generally corresponds to that of Figs. 10A-B, but now with a smaller link capacity C=1024 Kbps. In case of smaller links, the calculation of the TN matrix as proposed by equation (25) tends to be more conservative. In this case, values of TN may be relatively small, and therefore the fact that the elements of TN are integers may constitute a problem. For example, TN_{ij} = 4 may result in a significantly smaller admissible region than TN_{ij}=5. This problem may be solved by using any of a number of conventional interpolation techniques, allowing the elements of TN to take real values.

It is important to understand that the preceding description is merely intended to serve as a framework for an understanding of the invention.

In certain situations, it is not necessary to use both the linear and non-linear admissible regions. In fact, if the delay requirements are loose or the link capacities are large enough, it may be sufficient to check whether the traffic mix is contained within the non-linear admissible region. In other circumstances, it may be sufficient to use a construction of multiple linear admissible regions. In effect, the intersection of multiple linear admissible regions generally defines a non-linear region (which is piece-wise linear).

By properly identifying linear and non-linear admissible regions, the invention can also be applied to other transport mechanisms including natural extensions and developments of the basic UTRAN concept. For other types of multi-service networks, the non-linear admissible region and/or the linear admissible region or regions have to be identified according to the network-specific traffic characteristics and QoS requirements.

Notes on use with priority scheduling

It should be noted that the traffic delay also depends on the scheduling principle applied in the network. If packets of all services wait in the same queue (FIFO) and the packets are served in order of arrival, the most stringent delay requirement has to be met. This can be avoided by service differentiation, having different queues for services with different delay requirements. The delay-limited linear regions are equivalent to "effective bandwidths" being assigned to connections. Effective bandwidths calculated for FIFO scheduling can be extended directly for priority scheduling as proposed in reference [11]. With respect to the invention, this means that instead of a single linear region for each service class with FIFO scheduling, there will generally be multiple linear regions for each service class, depending on the number priority levels.

Prioritization means that a packet from a lower priority queue can be served only if all the higher priority queues are empty. Segmentation is used to minimize the influence of large low priority packets already in the server on high priority traffic. The segment size s is an additional model parameter. W_i and S_i apply to the last segment instead of the whole packet. W_i cannot be calculated directly using the work-load of the system because higher priority packets can over-take lower priority packets. Similarly to the FIFO case, the delay violation events are lost and delayed packets. When calculating ε_i^{lost} and $\varepsilon_i^{delayed}$, the only difference from the FIFO case is that from class-i point of view, the system is overloaded only if the input rate of traffic classes with higher (or equal) priority is higher than C.

We consider three cases depending on the priority levels of class-i and class-j. If class-i and class-j have the same priority level, then TN_{ij} is calculated in the same way as for FIFO scheduling set forth above. If class-i has higher priority we follow a different method. The event that the last segment (of size s_{last}) of the class-j packet could not be served before time D is equivalent to the event that all segments before the last one could not be served before $D' = D - s_{last}/C$. Formally, denote $B^{(i)}(0, t)$ the server availability in $[0, t]$ seen by the class-j packet, if the higher priority traffic is given by class-i, when it has arrived at time 0, and then:

$$\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) = \Pr\left(B^{(i)}(0, \widetilde{D}_j - \frac{s_{last}}{C}) < \frac{b_j - s_{last}}{C} \right). \tag{34}$$

By using a conservative approximation of the server availability process: $B^{(i)}(0, t) \approx t - A_i(0, t)/C$ and the Brownian bridge approximation of the arrival process, the following formula is obtained:

$$\begin{aligned} &\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) \approx \\ &\approx \Phi\left(b_j - s_{last}; (C - N_i\rho_i)D', N_i\rho_i^2 D'(TTI_i - D')\right), \end{aligned} \tag{35}$$

where $\Phi(x;\ \mu,\ \sigma^2)$ denotes the normal distribution. In order to demonstrate the accuracy of this approach, TN_{ij} values calculated using equation (35) are compared with exact values (C = 920 Kbps, b_i = 320 bits, s_{last} = 320 bits, TTI_i = 20ms, $\tilde{D}_j$= 10ms, $\tilde{\varepsilon}_j^{delayed}$ =0.1%) as illustrated below.

b_j [bits]	320	640	960	1920	2880	3840
TN_{ij} (exact)	41	38	36	30	25	20
TN_{ij} (approx.)	37	36	34	29	24	20

If class-i has lower priority, the effect of a segment possibly under service from class-i on delays of class-j packets is neglected. This means that the TN_{ij} values are set to infinity (∞).

In general, simulations have shown that priority scheduling is more advantageous than FIFO scheduling and the effect of the service differentiation is that the QoS requirements can be met at higher resource utilization.

Notes on use in the multiple-links scenario

The CAC algorithms proposed by the invention method has mainly been presented and evaluated for the single link scenario. In the multiple-links scenario, the overall CAC decision is composed of more than one Link Admission Control (LAC) decision. In practice, a method working in the multiple-links scenario is usually identical to the single-link algorithm, because no information on the resources along the end-to-end path is available. If the proposed methods are applied in the multiple-links scenario, an “overload-limited” region can be computed for the different links individually. Single-link effective bandwidths can be extended to the network level, e.g. as proposed in reference [12]. Essentially, the "effective bandwidths" calculated by the invention do not change in other links of the network. This means that the proposed single-link methods can be applied to the multiple-links scenario without modifications.

The embodiments described above are merely given as examples, and it should be understood that the present invention is not limited thereto. Further modifications, changes and improvements which retain the basic underlying principles disclosed and claimed herein are within the scope and spirit of the invention.

REFERENCES

[1] *Comparison of Call Admission Control Algorithms in ATM/AAL2 Based 3rd Generation Mobile Access Networks* by G. Fodor, G. Leijonhufvud, Sz. Malomsoky and A. Rácz, Proc. IEEE Wireless Communications and Networking Conference, 1999.

[2] *Connection Admission Control Design for GlobeView-2000 ATM Core Switches* by L. He and A. K. Wong, Bell Labs Technical Journal, pp. 94-110, January-March 1998.

[3] *Bounding On-Off Sources – Variability Ordering and Majorization to the Rescue* by A. M. Makowski, ISR TR 2001-13.

[4] 3GPP. *Synchronisation in UTRAN (Stage 2)*, Technical Specification, TR 25.402 V4.1.0, June, 2001.

[5] *Performance Evaluation and Dimensioning for AAL2 CLAD* by H. Saito, Proc. IEEE INFOCOM, pp. 153-160, 1999.

[6] 3GPP. *Delay Budget within the Access Stratum*, Technical Report TR 25.853 V4.0.0, May, 2001.

[7] *The Superposition of Variable Bit Rate Sources in an ATM Multiplexer* by Ilkka Norros, James W. Roberts, Alain Simonian, and Jorma T. Virtamo, IEEE Journal on Selected Areas in Communications, Vol. 9, No.3, pp. 378-387, 1991.

[8] *Methods for the performance evaluation and design of broadband multiservice networks, Part III, Traffic models and queuing analysis*, COST 242 Final Report, 1996.

[9] *A Queue with Periodic Arrivals and Constant Service Rate*, by B. Hayek, Probability, Statistics, and Optimisation, a Tribute to Peter Whittle, Wiley, pp. 147-157, 1994.

[10] *Notes on Effective Bandwidths*, by F. P. Kelly, Stochastic Networks: Theory and Applications, Vol. 4, Oxford University Press, pp. 141-168, 1996.

[11] *Effective Bandwidths with Priorities* by Arthur W. Berger and Ward Whitt, IEE/ACM Transactions on Networking, Vol. 6, No. 4, August 1998.

[12] The Output of a Switch, or Effective Bandwidths for Networks by Damon Wischik, Queuing Systems, Vol. 32, pp. 383-396, 1999.

APPENDIX A

An example of a software implementation

```
Global variables
  Capacity_kbps as Double

  nTTIs as Integer
  TTIs(1 To nTTIs) as Integer

  nClasses as Integer
  nClassesMax as Integer

  TETable(1 To nTTIs,1 To nClassesMax,1 To nClassesMax) as Double
  TNTable(1 To nTTIs,1 To nClassesMax) as Double

  NSourcesMax as Integer

  nActivities as Integer
  nActivitiesMax as Integer
  consideredActivities(1 To nActivitiesMax) as Double
  nMaxActive as Integer
  ActTable_1(1 To nActivitiesMax, 1 To NSourcesMax, 1 To nMaxActive) as Integer
  ActTable_2(1 To nActivitiesMax, 1 To NSourcesMax, 1 To nMaxActive) as Integer

  NSources(1 To nClassesMax) as Integer
  TTI_ms(1 To nClassesMax) as Double
  PacketSize_bit(1 To nClassesMax) as Double
  Activity(1 To nClassesMax) as Double
  Delay(1 To nClassesMax) as Double

  ActivityIndex(1 To nClassesMax) as Integer
  TTIIndexes(1 To nClassesMax) as Integer

  Loss as Double
  LossNDD as Double
  LossOnOff as Double

  nOnOffClasses as Integer
  TTIIndex as Integer
End Global variables
# ----------------------------------------------------------------------------------------------------
```

```
NewCall(_classIndex) As Boolean
  NSources(_classIndex) = NSources(_classIndex) + 1
  If NSources(_classIndex) = 1 Then
    ClassOn(_classIndex)
  End If

  Result=Admit()

  If Result=FALSE Then
    NSources(_classIndex) = NSources(_classIndex) - 1
    If NSources(_classIndex) = 0 Then
      ClassOff(_classIndex)
    End If
  End If

  Return Result
End
# ------------------------------------------------------------------------------------------------------------

EndCall(_classIndex)
  NSources(i)=NSources(i)-1
  If NSources(i)=0 Then
    ClassOff(_classIndex)
  End If
End
# ---------------------------------------------------------------------------

Admit()
  If AdmitNDD()=FALSE Then
    Return FALSE
  Else
    Return AdmitOnOff()
  End
End
# ---------------------------------------------------------------------------

AdmitNDD()
  For j = 1 To nClasses
    If NSources(j) > 0 Then
      Sum = 0
      For i = 1 To nClasses
        Sum = Sum + NSources(i) * TETable(TTIIndex, i, j)
      Next i
      If Sum > TNTable(TTIIndex, j)+1 Then
        Return FALSE
      End If
    End If
  Next j
  Return TRUE
End
# ----------------------------------------------------------------------------------------------------

AdmitOnOff()
  For i=1 To nClasses
    sumBw=0
    For j=1 To nClasses
      If ActivityIndex(j)=-1 OR NSources(j)=0 Then
        A=NSources(j)
      Else If i=j Then
        A = ActTable_2(ActivityIndex(j),NSources(j),nOnOffClasses)
      Else
        A = ActTable_1(ActivityIndex(j),NSources(j),nOnOffClasses)
      End If
      sumBw=sumBw + A * PacketSize_bit(j) / TTI_ms(j)
    Next j
    If sumBw > Capacity_kbps Then
      Return FALSE
    End If
  Next i
  Return TRUE
End
# ----------------------------------------------------------------------------------------------------

ClassOn(_classIndex)
  If TTIIndexes(_classIndex) > TTIIndex Then
    TTIIndex=TTIIndexes(_classIndex)
  End If
  If Activity(_classIndex) < 1 Then
    nOnOffClasses = nOnOffClasses + 1
  End If
End
# ----------------------------------------------------------------------------------------------------

ClassOff(_classIndex)
  If TTIIndexes(_classIndex) = TTIIndex Then
    TTIIndex=1
    For i=1 To nClasses
```

```
      If NSources(i) > 0 AND TTIIndexes(i) > TTIIndex Then
        TTIIndex=TTIIndexes(i)
      End If
    Next i
  End If

  If Activity(_classIndex) < 1 Then
    nOnOffClasses = nOnOffClasses - 1
  End If
End
# ----------------------------------------------------------------------------------------------------

Init()
    #set the following Values
  Capacity_kbps =
  nTTIs =
  TTIs() = #must be ordered
  nClassesMax=
  nActivitiesMax =
  nMaxActive =
  NSourcesMax =
  Loss=

    #defaults
  nClasses=0
  nActivities=0
  nOnOffClasses=0
  TTIIndex=1

    # can be different
  LossNDD=Loss/2
  LossOnOff=Loss-LossNDD
End
# ----------------------------------------------------------------------------------------------------

AddClassToTable(_TTI,_PacketSize,_Activity,_Delay)
  nClasses=nClasses+1 #not higher than nClassesMax
  classIndex=nClasses
  NSources(classIndex)=0

  TTI_ms(classIndex)=_TTI
  PacketSize_bit(classIndex)=_PacketSize
  Delay(classIndex)=_Delay
  Activity(classIndex)=_Activity
  If _Activity < 1 Then
    For i=1 To nActivities
      If consideredActivities(i)=_Activity Then
        Break
      End If
    Next i
    ActivityIndex(classIndex)=i
    If i>nActivities Then
      AddActivity(_Activity)
    End If
  Else
    ActivityIndex(classIndex)=-1
  End If

  For i=1 To nTTIs
    If TTIs(i)>= _TTI Then
      Break
    End If
  Next i
  TTIIndexes(classIndex)=min(i,nTTIs)

  SetTNs(classIndex)

  For j=1 To nClasses -1
    SetTEs(classIndex,j)
    SetTEs(j,classIndex)
  Next j
  SetTEs(classIndex,classIndex)
End
# -------------------------------------------------------------------------------------------------

AddActivity(_activity)
  nActivities=nActivities+1 #not higher than nActivitiesMax
  index=nActivities
  consideredActivities(index)=_activity
  For i=1 To NSourcesMax
    For j=1 To nMaxActive
      ActTable_1(index,i,j)=N_OnOff(i,_activity,LossOnOff/j,1)
      ActTable_2(index,i,j)=N_OnOff(i,_activity,LossOnOff/j,2)
    Next j
  Next i
End
# -------------------------------------------------------------------------------------------------
SetTNs(_classIndex)
```

```
  For i=TTIIndexes(_classIndex) To nTTIs
    TNTable(i,_classIndex) = CalcTN(_classIndex,_classIndex,TTIs(i))
    If PacketSize_bit(_classIndex) / Capacity_kbps < Delay(_classIndex) Then
        TNTable(i,_classIndex) = CAC_TNTable(i,_classIndex) + 1
    End If
  Next i
End
# -----------------------------------------------------------------------------------------------

SetTEs(_i,_j)
  For i=max(TTIIndexes(_i),TTIIndexes(_j)) To nTTIs
    If _i=_j Then
      TETable(i,_i,_j) = 1
    Else
      TETable(i,_i,_j) = TNTable(i,_j) / CalcTN(_i,_j,TTIs(i))
    End If
  Next i
End
# -----------------------------------------------------------------------------------------------

N_OnOff(N,p,x,mode)
#mode 1: BinomQx(N,P,x)
#quantile function of binomial distribution
#mode 2: quantile of weighted binomial distribution

CalcTN(_i,_j,_TTIMax)
#uses:  Capacity_kbps
#       TTI_ms
#       PacketSize_bit
#       Activity
#       Delay
#       LossNDD
```

What is claimed is:

1. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear admissible region;
- checking whether said traffic mix is contained also within at least one linear admissible region; and
- admitting said new connection for transport over said transport link only if said traffic mix is contained within an intersection of said at least one linear admissible region and said non-linear admissible region.

2. The method according to claim 1, wherein said admissible regions are related to respective quality of service (QoS) requirements.

3. The method according to claim 2, wherein said at least one linear admissible region contains a set of multi-service traffic mixes that fulfil a quality of service (QoS) requirement related to delay, and said non-linear admissible region contains a set of traffic mixes that fulfil a QoS requirement related to overload.

4. The method according to claim 1, wherein said non-linear admissible region is an overload-limited admissible region.

5. The method according to claim 4, wherein said overload-limited admissible region contains the set of traffic mixes for which the probability of temporarily overloading a queuing system associated with the transport link is smaller than a given target value.

6. The method according to claim 4, wherein said step of checking whether said traffic mix is contained within said non-linear overload-limited admissible region is

representative of checking whether or not said traffic mix violates a delay requirement related to packet loss caused by temporary overload of said transport link.

7. The method according to claim 4, wherein said step of checking whether said traffic mix is contained within said non-linear overload-limited admissible region comprises the step of evaluating the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link.

8. The method according to claim 7, wherein the per-class limit A_i is the number of connections from class-i such that the probability that more than A_i connections from class-i are active at the same time is smaller than a given target value.

9. The method according to claim 8, further comprising the steps of:

- pre-calculating at least some of said A_i values for a range of different values of the number N_i of connections from class-i or for a range of different activity factors α_i;
- storing said pre-calculated A_i values in memory; and
- accessing said pre-calculated A_i values from said memory for on-line evaluation of said inequalities.

10. The method according to claim 8, further comprising the step of determining A_i values by class-wise overload probability evaluation.

11. The method according to claim 10, wherein said step of determining A_i values comprises the step of finding minimal values of A_i such that the following sets of inequalities:

$$\sqrt[K_a]{1-\widetilde{\varepsilon}_i^{lost}} \geq \frac{\sum_{n_i=0}^{A_i} n_i \prod_i (n_i)}{N_i \alpha_i},$$

$$\sqrt[K_a]{1-\widetilde{\varepsilon}_i^{lost}} \geq \sum_{n_l=0}^{A_l} \prod_l (n_l) \qquad l = 1,2,\ldots,K_a, l \neq i,$$

are fulfilled, where K_a is the number of classes with activity factor α_i<1, $\widetilde{\varepsilon}_i^{lost}$ is the target packet loss probability for service class-i approximated by the target overload probability assigned to class-i, N_i is the number of connections from class-i and n_i is the number of actually active connections from class-i.

12. The method according to claim 1, wherein said at least one linear admissible region is in the form of at least one delay-limited admissible region.

13. The method according to claim 12, wherein said at least one delay-limited admissible region is in the form of multiple class-specific delay-limited admissible regions.

14. The method according to claim 13, wherein each class-specific delay-limited admissible region is a linear admissible region that contains a set of traffic mixes fulfilling a given class-specific packet delay requirement.

15. The method according to claim 14, wherein said class-specific packet delay requirement requires that the probability of the class-specific packet delay being larger than a given class-specific maximum delay is smaller than a given target value.

16. The method according to claim 13, wherein said step of checking whether said traffic mix is contained also within multiple class-specific, delay-limited admissible regions comprises the step of evaluating the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix.

17. The method according to claim 16, wherein TE_{ij} is calculated in the following way:

$TE_{ij} = TN_{jj}/\ TN_{ij}$, and

TN_{ij} is calculated in the following way:

$$TN_{ij} = \max\left\{ N_i \,\middle|\, \sum_{n_i=0}^{N_i} \Pi(n_i)\ \Pr\left(D_j^{(i)} > \widetilde{D}_j \middle| n_i \text{ connections are active}\right) \leq \widetilde{\varepsilon}_j^{\,delayed} \right\},$$

where $D_j^{(i)}$ denotes the delay of a packet from class-j assuming that the delay of the associated queue comes from only class-i connections, $\widetilde{D}_j$ is the target delay criteria of packets from class-j, $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is the probability of packet delay criteria violation, $\widetilde{\varepsilon}_j^{\,delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost and n_i is the number of actually active connections from class-i.

18. The method according to claim 17, wherein the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is calculated in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) =$$

$$= \sum_{x'<l\leq n_i} \binom{n_i}{l} \left(\frac{l-x'}{TTI'}\right)^{l} \left(1-\frac{l-x'}{TTI'}\right)^{n_i-l} \cdot \frac{TTI'-n_i+x'}{TTI'-l+x'}$$

$$x' = (\widetilde{D}_j - \frac{b_j}{C}) / TU$$

$$TTI' = TTI_i / TU$$

$$TU = \frac{b_i}{C},$$

where b_j is the class-j packet size, C is the capacity of said transport link and TTI_i is the relevant packet inter-arrival time.

19. The method according to claim 17, wherein the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is calculated in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j \,|\, n_i \text{ connections are active}) = \exp\left\{ -\frac{2Cx}{TTI_i\, n_i\, \rho_i^2} \left(\frac{Cx}{TTI_i} + C - n_i\, \rho_i \right) \right\}$$

$$x = \widetilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, TTI_i is the relevant packet inter-arrival time, b_j is the class-j packet size and ρ_i is the average load generated by one active traffic source from class-i.

20. The method according to claim 16, wherein TE_{ij} is defined as $TE_{ij} = TN_{jj}/\ TN_{ij}$, and TN_{ij} is calculated in the following way:

$$TN_{ij} = \left\lceil \frac{C + \dfrac{C x \alpha_i}{TTI_i}}{\alpha_i \rho_i - \dfrac{\alpha_i \rho_i^2 \, TTI_i \ln(\tilde{\varepsilon}_j^{delayed})}{2x}} \right\rceil$$

$$x = \tilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, α_i is the activity factor of class-i, TTI_i is the relevant packet inter-arrival time, ρ_i is the average load generated by one active traffic source from class-i, b_j is the class-j packet size and $\tilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost.

21. The method according to claim 16, further comprising the step of updating TN_{ij} and TE_{ij}, before said step of checking whether said traffic mix is contained within said intersection of admissible regions, only when said new connection belongs to a new service class.

22. The method according to claim 16, further comprising the step of assigning TN_{ij} a real value by means of interpolation.

23. The method according to claim 1, wherein said communication network is a transport network based on the Universal Terrestrial Radio Access Network (UTRAN).

24. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear overload-limited admissible region; and
- admitting said new connection for transport over said transport link only if said traffic mix is contained within said non-linear overload-limited admissible region.

25. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an intersection of multiple service-class-specific delay-limited admissible regions; and
- admitting said new connection for transport over said transport link only if said traffic mix is contained within said intersection of admissible regions.

26. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear admissible region;
- means for checking whether said traffic mix is contained also within at least one linear admissible region; and
- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within an intersection of said at least one linear admissible region and said non-linear admissible region.

27. The admission controller according to claim 26, wherein said admissible regions are related to respective quality of service (QoS) requirements.

28. The admission controller according to claim 27, wherein said at least one linear admissible region contains a set of multi-service traffic mixes that fulfil a quality of service (QoS) requirement related to delay, and said non-linear admissible region contains a set of traffic mixes that fulfil a QoS requirement related to overload.

29. The admission controller according to claim 26, wherein said non-linear admissible region is an overload-limited admissible region.

30. The admission controller according to claim 29, wherein said overload-limited admissible region contains the set of traffic mixes for which the probability of temporarily overloading a queuing system associated with the transport link is smaller than a given target value.

31. The admission controller according to claim 29, wherein said means for checking whether said traffic mix is contained within said non-linear overload-limited admissible region is operable for checking whether said traffic mix violates a packet delay requirement related to packet loss caused by temporary overload of said transport link.

32. The admission controller according to claim 29, wherein said means for checking whether said traffic mix is contained within said non-linear overload-limited admissible region comprises means for evaluating the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C ,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link.

33. The admission controller according to claim 32, wherein the per-class limit A_i is the number of connections from class-i such that the probability that more than A_i connections from class-i are active at the same time is smaller than a given target value.

34. The admission controller according to claim 33, further comprising:

- means for pre-calculating at least some of said A_i values for a range of different values of the number N_i of connections from class-i or for a range of different activity factors α_i;
- means for storing said pre-calculated A_i values in memory; and
- means for accessing said pre-calculated A_i values from said memory for on-line evaluation of said inequalities.

35. The admission controller according to claim 33, further comprising means for determining A_i values by class-wise overload probability evaluation.

36. The admission controller according to claim 35, wherein said means for determining A_i values comprises means for finding minimal values of A_i such that the following sets of inequalities:

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \frac{\sum_{n_i=0}^{A_i} n_i \prod_i (n_i)}{N_i \alpha_i},$$

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \sum_{n_l=0}^{A_l} \prod_l (n_l) \qquad l=1,2,\ldots,K_a, l \neq i,$$

are fulfilled, where K_a is the number of service classes with activity factor α_i<1, $\tilde{\varepsilon}_i^{lost}$ is the target packet loss probability for service class-i approximated by the target overload probability assigned to class-i, N_i is the number of connections from class-i and n_i is the number of actually active connections from class-i.

37. The admission controller according to claim 26, wherein said at least one linear admissible region is in the form of at least one delay-limited admissible region.

38. The admission controller according to claim 37, wherein said at least one delay-limited admissible region is in the form of multiple class-specific delay-limited admissible regions.

39. The admission controller according to claim 38, wherein each class-specific delay-limited admissible region is a linear admissible region that contains a set of traffic mixes fulfilling a given class-specific packet delay requirement.

40. The admission controller according to claim 39, wherein said class-specific packet delay requirement requires that the probability of the class-specific packet delay being larger than a given class-specific maximum delay is smaller than a given target value.

41. The admission controller according to claim 38, wherein said means for checking whether said traffic mix is contained also within multiple class-specific, delay-limited admissible regions comprises means for evaluating the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix.

42. The admission controller according to claim 41, wherein said means for checking whether said traffic mix is contained also within multiple class-specific, delay-limited admissible regions comprises:

- means for calculating TE_{ij} in the following way:

$$TE_{ij} = TN_{jj}/\ TN_{ij};\ \text{and}$$

- means for calculating TN_{ij} in the following way:

$$TN_{ij} = \max\left\{ N_i \;\middle|\; \sum_{n_i=0}^{N_i} \Pi(n_i)\ \Pr\left(D_j^{(i)} > \tilde{D}_j \middle| n_i \text{ connections are active}\right) \le \tilde{\varepsilon}_j^{delayed} \right\},$$

where $D_j^{(i)}$ denotes the delay of a packet from class-j assuming that the delay of the associated queue comes from only class-i connections, $\tilde{D}_j$ is the target delay criteria of packets from class-j, $\Pr(D_j^{(i)} > \tilde{D}_j | n_i \text{ connections are active})$ is the probability of packet delay criteria violation, $\tilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost and n_i is the number of actually active connections from class-i.

43. The admission controller according to claim 42, wherein said means for calculating TN_{ij} comprises means for calculating the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \tilde{D}_j | n_i \text{ connections are active})$ in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) =$$

$$= \sum_{x'<l\leq n_i} \binom{n_i}{l} \left(\frac{l-x'}{TTI'}\right)^l \left(1-\frac{l-x'}{TTI'}\right)^{n_i-l} \cdot \frac{TTI'-n_i+x'}{TTI'-l+x'}$$

$$x' = (\widetilde{D}_j - \frac{b_j}{C}) / TU$$

$$TTI' = TTI_i / TU$$

$$TU = \frac{b_i}{C},$$

where b_j is the class-j packet size, C is the capacity of said transport link and TTI_i is the relevant packet inter-arrival time.

44. The admission controller according to claim 42, wherein said means for calculating TN_{ij} comprises means for calculating the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j \,|\, n_i \text{ connections are active}) = \exp\left\{-\frac{2Cx}{TTI_i\, n_i\, \rho_i^2}\left(\frac{Cx}{TTI_i} + C - n_i\,\rho_i\right)\right\}$$

$$x = \widetilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, TTI_i is the relevant packet inter-arrival time, b_j is the class-j packet size and ρ_i is the average load generated by one active traffic source from class-i.

45. The admission controller according to claim 41, wherein said means for checking whether said traffic mix is contained also within multiple class-specific, delay-limited admissible regions comprises:

- means for calculating TE_{ij} in the following way:

$$TE_{ij} = TN_{jj} / TN_{ij}; \text{ and}$$

- means for calculating TN_{ij} in the following way:

$$TN_{ij} = \left\lceil \frac{C + \dfrac{C x \alpha_i}{TTI_i}}{\alpha_i \rho_i - \dfrac{\alpha_i \rho_i^2 \, TTI_i \ln(\tilde{\varepsilon}_j^{delayed})}{2x}} \right\rceil$$

$$x = \tilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, α_i is the activity factor of class-i, TTI_i is the relevant packet inter-arrival time, ρ_i is the average load generated by one active traffic source from class-i, b_j is the class-j packet size and $\tilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost.

46. The admission controller according to claim 41, further comprising means for updating TN_{ij} and TE_{ij}, before checking whether said traffic mix is contained within said intersection of admissible regions, when said new connection belongs to a new service class.

47. The admission controller according to claim 41, further comprising means for assigning TN_{ij} a real value by means of interpolation.

48. The admission controller according to claim 26, wherein said communication network is a transport network based on the Universal Terrestrial Radio Access Network (UTRAN).

49. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear overload-limited admissible region; and
- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within said non-linear overload-limited admissible region.

50. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an intersection of multiple service-class-specific delay-limited admissible regions; and
- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within said intersection of admissible regions.

AMENDED CLAIMS

[received by the International Bureau on 24 February 2003 (24.02.03); original claims 1-50 replaced by new claims 1-42]

1. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service-class traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an overload-limited admissible region defined as a non-linear admissible region that contains a set of traffic mixes that fulfil a given overload requirement, where the dimensions of said non-linear admissible region are the number of connections in the respective service classes;
- checking, for each of a number of said service classes, whether said traffic mix is contained also within a class-specific delay-limited admissible region approximated as a linear admissible region that contains a set of traffic mixes that fulfil a given class-specific delay requirement, where the dimensions of said linear admissible region are the number of connections in the respective service classes; and
- admitting said new connection for transport over said transport link only if said traffic mix is contained within an intersection of said non-linear overload-limited admissible region and said linear delay-limited admissible region(s).

2. The method according to claim 1, wherein said delay-limited region is approximated as a linear region for a multi-service-class traffic mix generally modeled as a superposition of periodic on-off connections.

3. The method according to claim 1, wherein said overload-limited admissible region contains the set of traffic mixes for which the probability of temporarily overloading a queuing system associated with the transport link is smaller than a given target value.

4. The method according to claim 1, wherein said step of checking whether said traffic mix is contained within said non-linear overload-limited admissible region is

representative of checking whether or not said traffic mix violates a delay requirement related to packet loss caused by temporary overload of said transport link.

5. The method according to claim 1, wherein said step of checking whether said traffic mix is contained within said non-linear overload-limited admissible region comprises the step of evaluating the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link.

6. The method according to claim 5, wherein the per-class limit A_i is the number of connections from class-i such that the probability that more than A_i connections from class-i are active at the same time is smaller than a given target value.

7. The method according to claim 6, further comprising the steps of:

- pre-calculating at least some of said A_i values for a range of different values of the number N_i of connections from class-i or for a range of different activity factors α_i;
- storing said pre-calculated A_i values in memory; and
- accessing said pre-calculated A_i values from said memory for on-line evaluation of said inequalities.

8. The method according to claim 6, further comprising the step of determining A_i values by class-wise overload probability evaluation.

9. The method according to claim 8, wherein said step of determining A_i values comprises the step of finding values of A_i such that the following sets of inequalities:

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \frac{\sum_{n_i=0}^{A_i} n_i \prod_i(n_i)}{N_i \alpha_i},$$

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \sum_{n_l=0}^{A_l} \prod_l(n_l) \qquad l = 1,2,\ldots,K_a, l \neq i,$$

are fulfilled, where K_a is the number of classes with activity factor α_i<1, $\tilde{\varepsilon}_i^{lost}$ is the target packet loss probability for service class-i approximated by the target overload probability assigned to class-i, N_i is the number of connections from class-i and n_i is the number of actually active connections from class-i.

10. The method according to claim 1, wherein said class-specific packet delay requirement requires that the probability of the class-specific packet delay being larger than a given class-specific maximum delay is smaller than a given target value.

11. The method according to claim 1, comprising the step of checking whether said traffic mix is contained within multiple class-specific, delay-limited admissible regions by evaluating the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix.

12. The method according to claim 11, wherein TE_{ij} is calculated in the following way:

$$TE_{ij} = TN_{jj} / TN_{ij}, \text{ and}$$

TN_{ij} is calculated in the following way:

$$TN_{ij} = \max\left\{ N_i \,\middle|\, \sum_{n_i=0}^{N_i} \Pi(n_i) \ \Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) \le \widetilde{\varepsilon}_j^{delayed} \right\},$$

where $D_j^{(i)}$ denotes the delay of a packet from class-j assuming that the delay of the associated queue comes from only class-i connections, $\widetilde{D}_j$ is the target delay criteria of packets from class-j, $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is the probability of packet delay criteria violation, $\widetilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost and n_i is the number of actually active connections from class-i.

13. The method according to claim 12, wherein the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is calculated in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) =$$

$$= \sum_{x' < l \le n_i} \binom{n_i}{l} \left(\frac{l - x'}{TTI'} \right)^l \left(1 - \frac{l - x'}{TTI'} \right)^{n_i - l} \cdot \frac{TTI' - n_i + x'}{TTI' - l + x'}$$

$$x' = (\widetilde{D}_j - \frac{b_j}{C}) / TU$$

$$TTI' = TTI_i / TU$$

$$TU = \frac{b_i}{C},$$

where b_j is the class-j packet size, C is the capacity of said transport link and TTI_i is the relevant packet inter-arrival time.

14. The method according to claim 12, wherein the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is calculated in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j \,|\, n_i \text{ connections are active}) = \exp\left\{-\frac{2Cx}{TTI_i\, n_i\, \rho_i^2}\left(\frac{Cx}{TTI_i} + C - n_i\, \rho_i\right)\right\}$$

$$x = \widetilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, TTI_i is the relevant packet inter-arrival time, b_j is the class-j packet size and ρ_i is the average load generated by one active traffic source from class-i.

15. The method according to claim 11, wherein TE_{ij} is defined as $TE_{ij} = TN_{jj} / TN_{ij}$, and TN_{ij} is calculated in the following way:

$$TN_{ij} = \left\lceil \frac{C + \dfrac{Cx\alpha_i}{TTI_i}}{\alpha_i\, \rho_i - \dfrac{\alpha_i\, \rho_i^2\, TTI_i \ln(\widetilde{\varepsilon}_j^{\,delayed})}{2x}} \right\rceil$$

$$x = \widetilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, α_i is the activity factor of class-i, TTI_i is the relevant packet inter-arrival time, ρ_i is the average load generated by one active traffic source from class-i, b_j is the class-j packet size and $\tilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost.

16. The method according to claim 11, further comprising the step of updating TN_{ij} and TE_{ij}, before said step of checking whether said traffic mix is contained within said intersection of admissible regions, only when said new connection belongs to a new service class.

17. The method according to claim 11, further comprising the step of assigning TN_{ij} a real value by means of interpolation.

18. The method according to claim 12, wherein, if class-i packets have higher priority than class-j packets, the probability of packet delay criteria violation is calculated in the following way:

$$\Pr\left(B^{(i)}(0,\widetilde{D}_j - \frac{s_{last}}{C}) < \frac{b_j - s_{last}}{C} \right),$$

where $B^{(i)}(0, t)$ denotes the server availability in $[0, t]$ seen by the class-j packet arriving at time 0, s_{last} denotes the size of the last segment of the class-j packet, and b_j is the class-j packet size.

19. The method according to claim 1, wherein said communication network is a transport network based on the Universal Terrestrial Radio Access Network (UTRAN).

20. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear overload-limited admissible region by evaluating the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link; and

- admitting said new connection for transport over said transport link only if said traffic mix is contained within said non-linear overload-limited admissible region.

21. A method for controlling admission of a new connection onto a transport link in a communication network, said method comprising the steps of:

- checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an intersection of multiple service-class-specific delay-limited admissible regions by evaluating the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix; and

- admitting said new connection for transport over said transport link only if said traffic mix is contained within said intersection of admissible regions.

22. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service-class traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an overload-limited admissible region defined as a non-linear admissible region that contains a set of traffic mixes that fulfil a given overload requirement, where the dimensions of said non-linear admissible region are the number of connections in the respective service classes;
- means for checking, for each of a number of said service classes, whether said traffic mix is contained also within a class-specific delay-limited admissible region approximated as a linear admissible region that contains a set of traffic mixes that fulfil a given class-specific delay requirement, where the dimensions of said linear admissible region are the number of connections in the respective service classes; and
- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within an intersection of said non-linear overload-limited admissible region and said linear delay-limited admissible region(s).

23. The admission controller according to claim 22, wherein said delay-limited region is approximated as a linear region for a multi-service-class traffic mix generally modeled as a superposition of periodic on-off connections.

24. The admission controller according to claim 22, wherein said overload-limited admissible region contains the set of traffic mixes for which the probability of

temporarily overloading a queuing system associated with the transport link is smaller than a given target value.

25. The admission controller according to claim 22, wherein said means for checking whether said traffic mix is contained within said non-linear overload-limited admissible region is operable for checking whether said traffic mix violates a packet delay requirement related to packet loss caused by temporary overload of said transport link.

26. The admission controller according to claim 22, wherein said means for checking whether said traffic mix is contained within said non-linear overload-limited admissible region comprises means for evaluating the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C ,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link.

27. The admission controller according to claim 26, wherein the per-class limit A_i is the number of connections from class-i such that the probability that more than A_i connections from class-i are active at the same time is smaller than a given target value.

28. The admission controller according to claim 27, further comprising:

- means for pre-calculating at least some of said A_i values for a range of different values of the number N_i of connections from class-i or for a range of different activity factors α_i;
- means for storing said pre-calculated A_i values in memory; and

- means for accessing said pre-calculated A_i values from said memory for on-line evaluation of said inequalities.

29. The admission controller according to claim 27, further comprising means for determining A_i values by class-wise overload probability evaluation.

30. The admission controller according to claim 29, wherein said means for determining A_i values comprises means for finding values of A_i such that the following sets of inequalities:

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \frac{\sum_{n_i=0}^{A_i} n_i \prod_i (n_i)}{N_i \alpha_i},$$

$$\sqrt[K_a]{1-\tilde{\varepsilon}_i^{lost}} \geq \sum_{n_l=0}^{A_l} \prod_l (n_l) \qquad l = 1,2,\ldots,K_a, l \neq i,$$

are fulfilled, where K_a is the number of service classes with activity factor α_i<1, $\tilde{\varepsilon}_i^{lost}$ is the target packet loss probability for service class-i approximated by the target overload probability assigned to class-i, N_i is the number of connections from class-i and n_i is the number of actually active connections from class-i.

31. The admission controller according to claim 22, wherein said class-specific packet delay requirement requires that the probability of the class-specific packet delay being larger than a given class-specific maximum delay is smaller than a given target value.

32. The admission controller according to claim 22, comprising means for checking whether said traffic mix is contained within multiple class-specific, delay-limited admissible regions based on evaluation of the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix.

33. The admission controller according to claim 32, wherein said means for checking whether said traffic mix is contained within multiple class-specific, delay-limited admissible regions comprises:

- means for calculating TE_{ij} in the following way:

$$TE_{ij} = TN_{jj} / \, TN_{ij}; \text{ and}$$

- means for calculating TN_{ij} in the following way:

$$TN_{ij} = \max\left\{ N_i \,\middle|\, \sum_{n_i=0}^{N_i} \Pi(n_i) \ \Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) \leq \widetilde{\varepsilon}_j^{delayed} \right\},$$

where $D_j^{(i)}$ denotes the delay of a packet from class-j assuming that the delay of the associated queue comes from only class-i connections, $\widetilde{D}_j$ is the target delay criteria of packets from class-j, $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ is the probability of packet delay criteria violation, $\widetilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost and n_i is the number of actually active connections from class-i.

34. The admission controller according to claim 33, wherein said means for calculating TN_{ij} comprises means for calculating the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active}) =$$

$$= \sum_{x'<l\le n_i} \binom{n_i}{l} \left(\frac{l-x'}{TTI'}\right)^l \left(1-\frac{l-x'}{TTI'}\right)^{n_i-l} \cdot \frac{TTI'-n_i+x'}{TTI'-l+x'}$$

$$x' = (\widetilde{D}_j - \frac{b_j}{C}) / TU$$

$$TTI' = TTI_i / TU$$

$$TU = \frac{b_i}{C},$$

where b_j is the class-j packet size, C is the capacity of said transport link and TTI_i is the relevant packet inter-arrival time.

35. The admission controller according to claim 33, wherein said means for calculating TN_{ij} comprises means for calculating the probability of packet delay criteria violation $\Pr(D_j^{(i)} > \widetilde{D}_j | n_i \text{ connections are active})$ in the following way:

$$\Pr(D_j^{(i)} > \widetilde{D}_j \,|\, n_i \text{ connections are active}) = \exp\left\{-\frac{2Cx}{TTI_i\, n_i\, \rho_i^2}\left(\frac{Cx}{TTI_i} + C - n_i\,\rho_i\right)\right\}$$

$$x = \widetilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, TTI_i is the relevant packet inter-arrival time, b_j is the class-j packet size and ρ_i is the average load generated by one active traffic source from class-i.

36. The admission controller according to claim 32, wherein said means for checking whether said traffic mix is contained also within multiple class-specific, delay-limited admissible regions comprises:

- means for calculating TE_{ij} in the following way:

$$TE_{ij} = TN_{jj} / TN_{ij}; \text{ and}$$

- means for calculating TN_{ij} in the following way:

$$TN_{ij} = \left\lceil \frac{C + \frac{C x \alpha_i}{TTI_i}}{\alpha_i \rho_i - \frac{\alpha_i \rho_i^2 \, TTI_i \ln(\tilde{\varepsilon}_j^{delayed})}{2x}} \right\rceil$$

$$x = \tilde{D}_j - \frac{b_j}{C},$$

where C is the capacity of said transport link, α_i is the activity factor of class-i, TTI_i is the relevant packet inter-arrival time, ρ_i is the average load generated by one active traffic source from class-i, b_j is the class-j packet size and $\tilde{\varepsilon}_j^{delayed}$ is the target value for the probability of a packet exceeding its delay criteria without getting lost.

37. The admission controller according to claim 32, further comprising means for updating TN_{ij} and TE_{ij}, before checking whether said traffic mix is contained within said intersection of admissible regions, when said new connection belongs to a new service class.

38. The admission controller according to claim 32, further comprising means for assigning TN_{ij} a real value by means of interpolation.

39. The admission controller according to claim 33, wherein, if class-i packets have higher priority than class-j packets, the probability of packet delay criteria violation is calculated in the following way:

$$\Pr\left(B^{(i)}(0,\widetilde{D}_j - \frac{s_{last}}{C}) < \frac{b_j - s_{last}}{C} \right),$$

where $B^{(i)}(0, t)$ denotes the server availability in $[0, t]$ seen by the class-j packet arriving at time 0, s_{last} denotes the size of the last segment of the class-j packet, and b_j is the class-j packet size.

40. The admission controller according to claim 22, wherein said communication network is a transport network based on the Universal Terrestrial Radio Access Network (UTRAN).

41. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within a non-linear overload-limited admissible region based on evaluation of the following inequalities:

$$\sum_{i=1}^{K} A_i \rho_i \leq C,$$

where K is the number of service classes in said traffic mix, A_i is a per-class limit on the number of simultaneously active connections, ρ_i is the average load generated by one active traffic source from class-i and C is the capacity of said transport link; and

- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within said non-linear overload-limited admissible region.

42. An admission controller for controlling admission of a new connection onto a transport link in a communication network, said admission controller comprising:

- means for checking whether a multi-service traffic mix defined by previously admitted connections present on said link together with said new connection is contained within an intersection of multiple service-class-specific delay-limited admissible regions based on evaluation of the following inequalities:

$$\sum_{i=1}^{K} N_i \cdot TE_{ij} \leq TN_{jj} + \text{constant}, \quad j = 1, 2, \ldots, K,$$

where K is the number of service classes in said traffic mix, TN_{ij} is a representation of the maximum number of connections from class-i assuming that a packet from class-j would fulfil a packet delay requirement of class-j, TE_{ij} is a service class equivalent measure representing how many new connections can be admitted from class-j in place of a connection from class-i considering only the packet delay requirement of class-j and N_i is the number of connections from class-i in the traffic mix; and

- means for admitting said new connection for transport over said transport link only if said traffic mix is contained within said intersection of admissible regions.

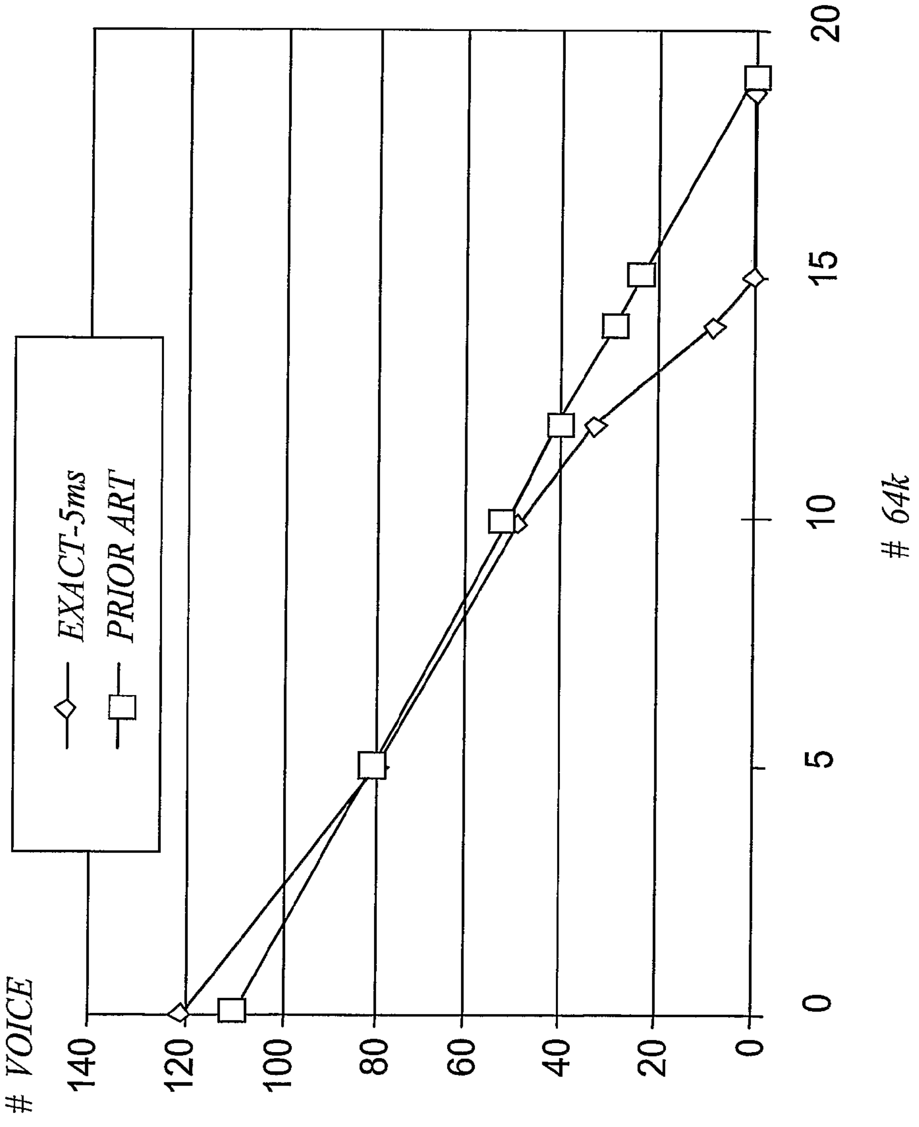
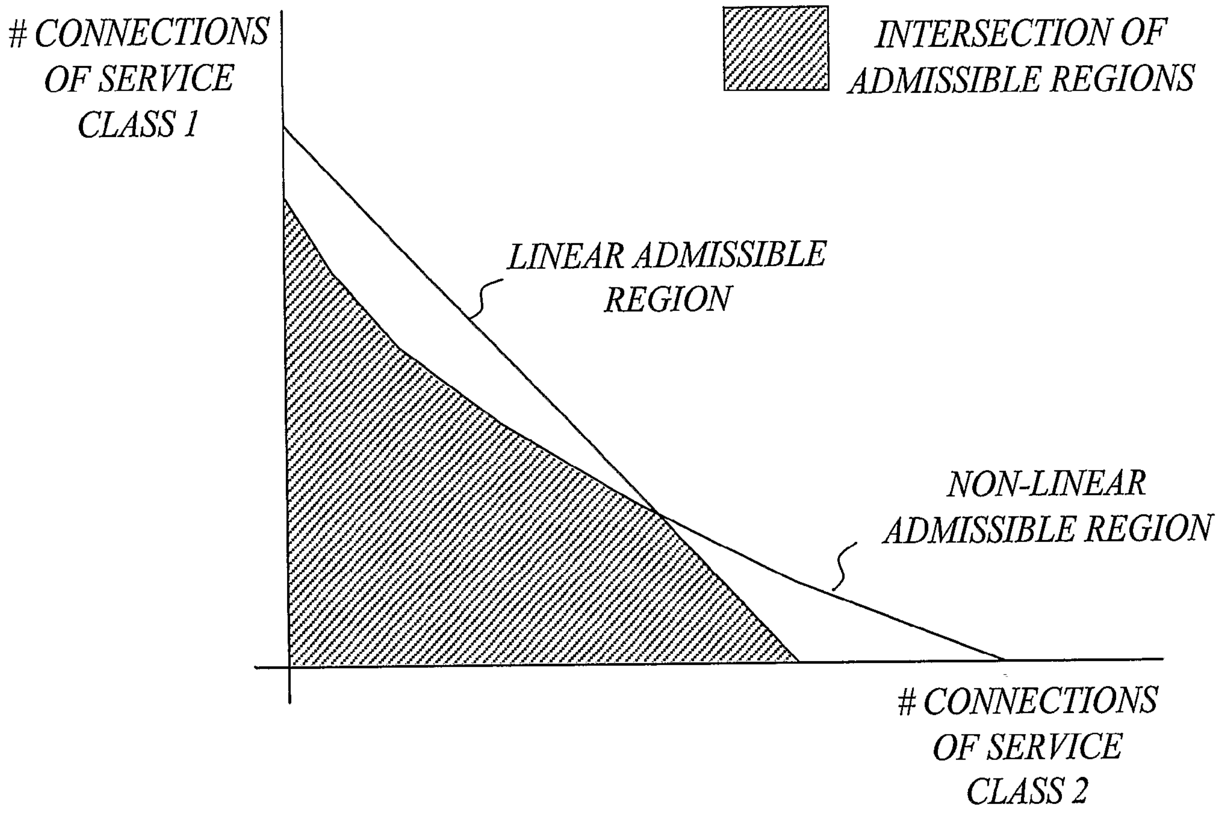

# VOICE
140
120
100
80
60
40
20
0
0
5
10
15
20
# 64k
EXACT-5ms
PRIOR ART

Fig. 1



# CONNECTIONS OF SERVICE CLASS 1
INTERSECTION OF ADMISSIBLE REGIONS
LINEAR ADMISSIBLE REGION
NON-LINEAR ADMISSIBLE REGION
# CONNECTIONS OF SERVICE CLASS 2

Fig. 2

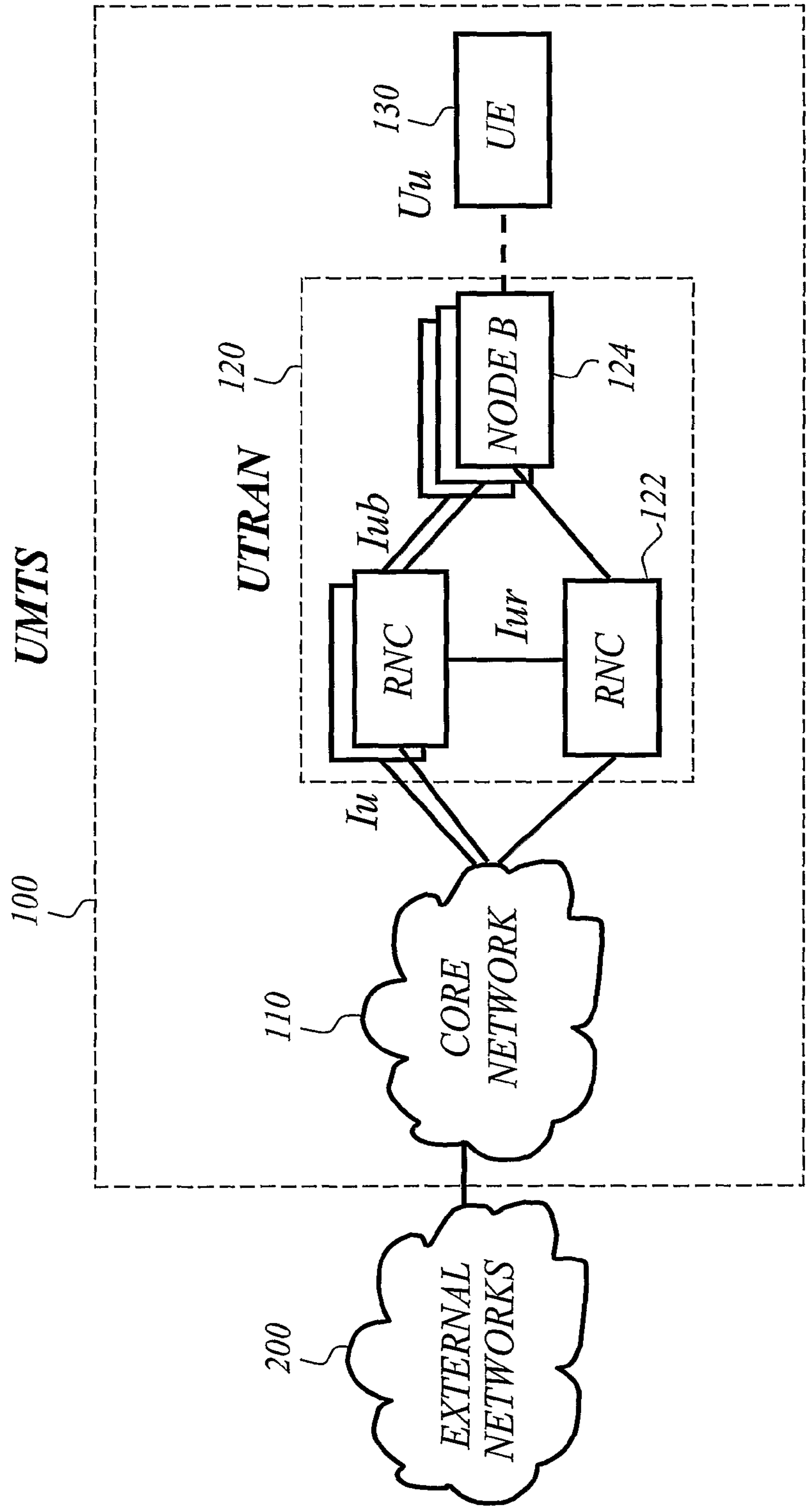

UMTS
100
UTRAN
120
UE
130
Uu
NODE B
124
Iub
RNC
Iur
RNC
122
Iu
CORE NETWORK
110
EXTERNAL NETWORKS
200

Fig. 3

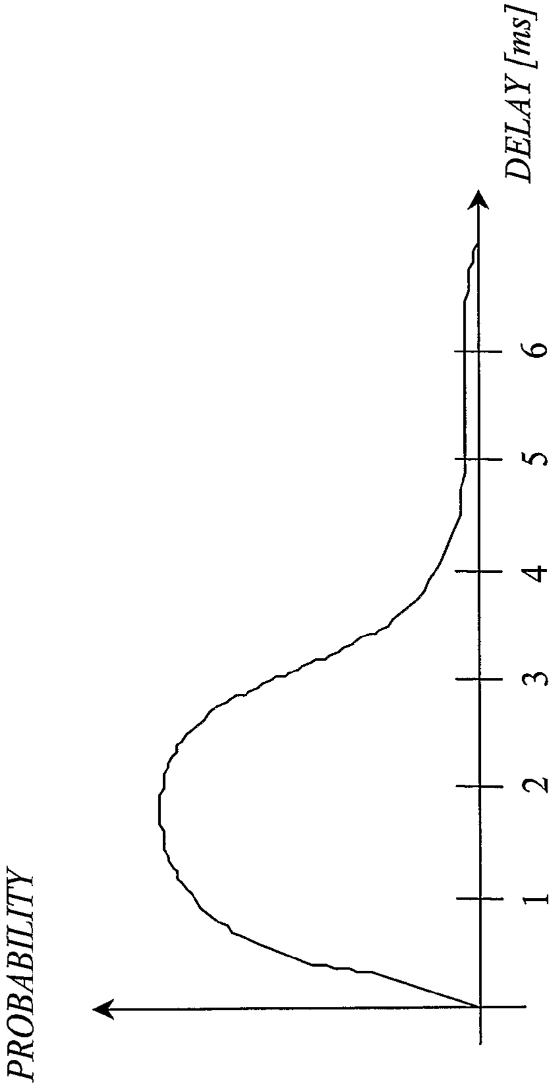


Fig. 4

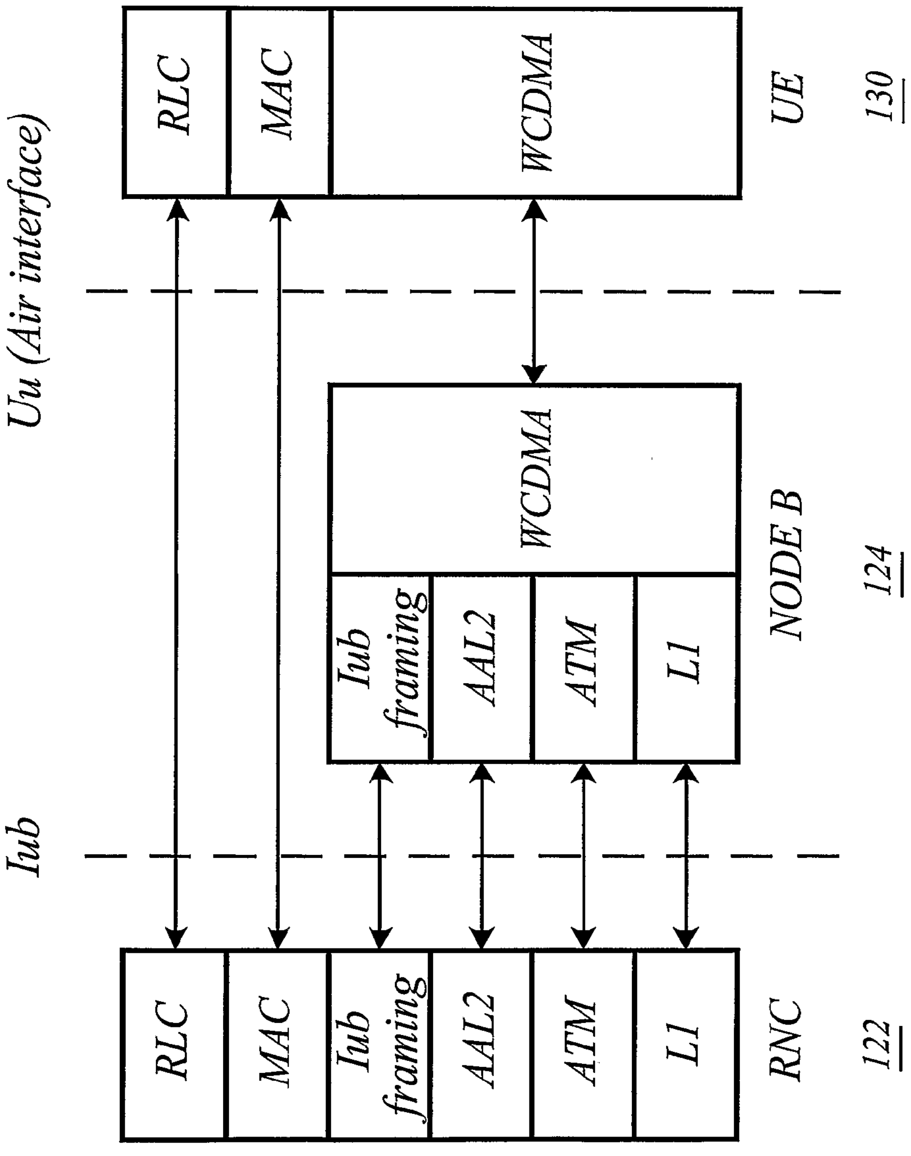

Uu (Air interface)
Iub
RLC
MAC
WCDMA
UE
130
WCDMA
Iub framing
AAL2
ATM
L1
NODE B
124
RLC
MAC
Iub framing
AAL2
ATM
L1
RNC
122

Fig. 5

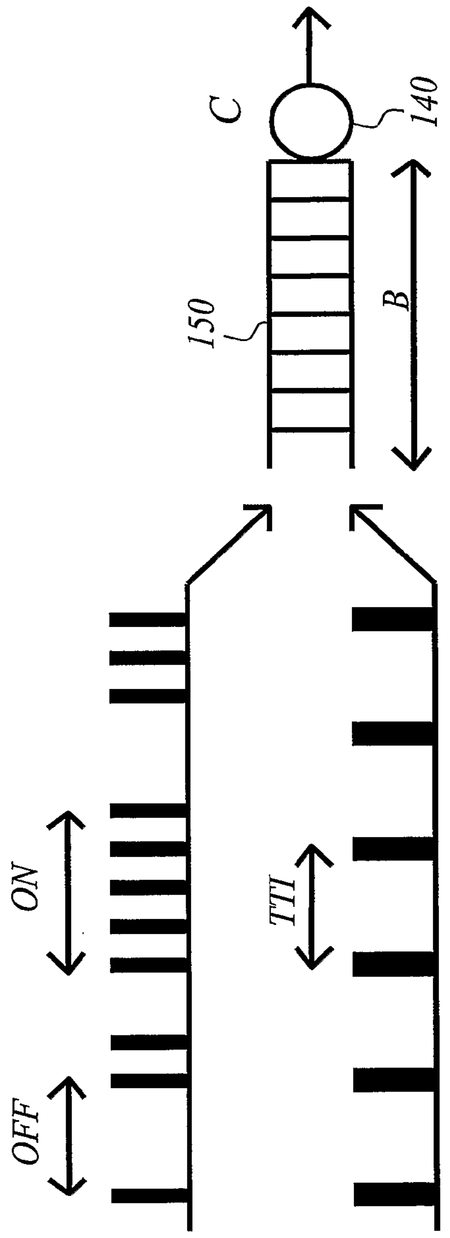


Fig. 6

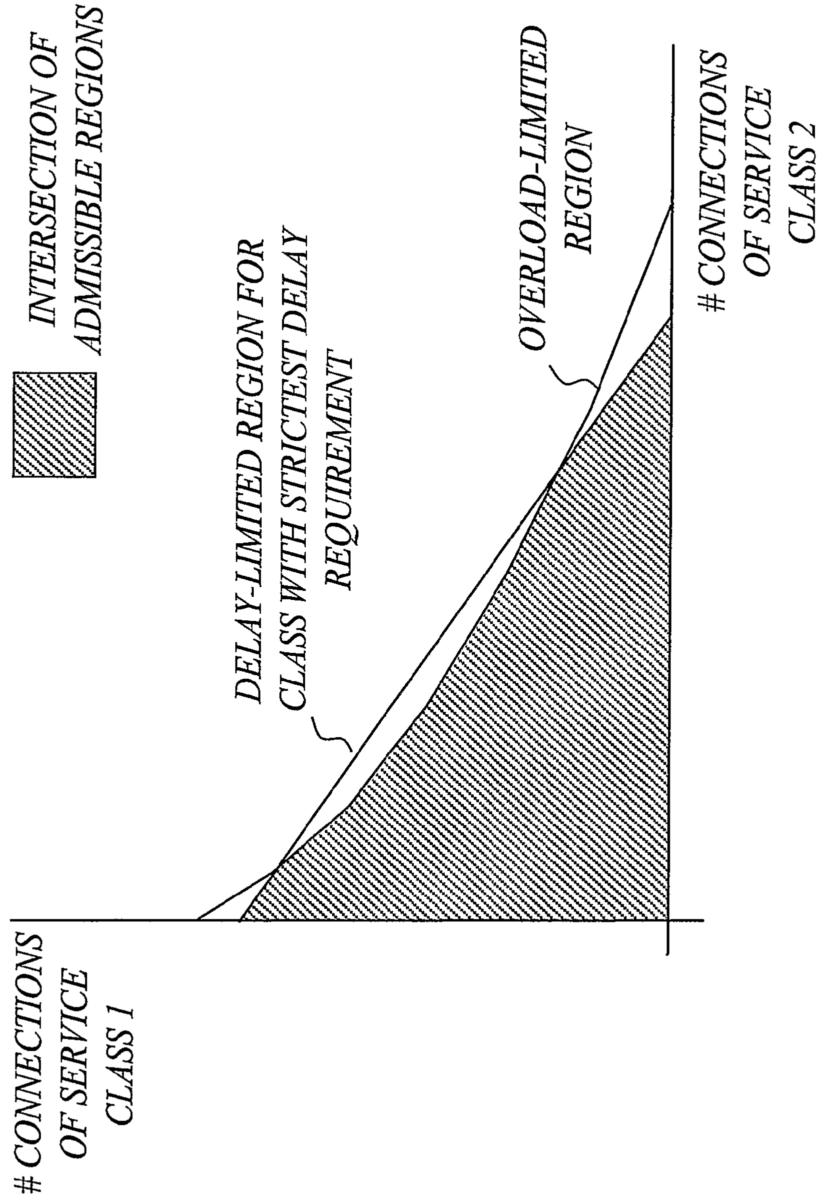

▨ INTERSECTION OF ADMISSIBLE REGIONS
DELAY-LIMITED REGION FOR CLASS WITH STRICTEST DELAY REQUIREMENT
OVERLOAD-LIMITED REGION
# CONNECTIONS OF SERVICE CLASS 1
# CONNECTIONS OF SERVICE CLASS 2

Fig. 7

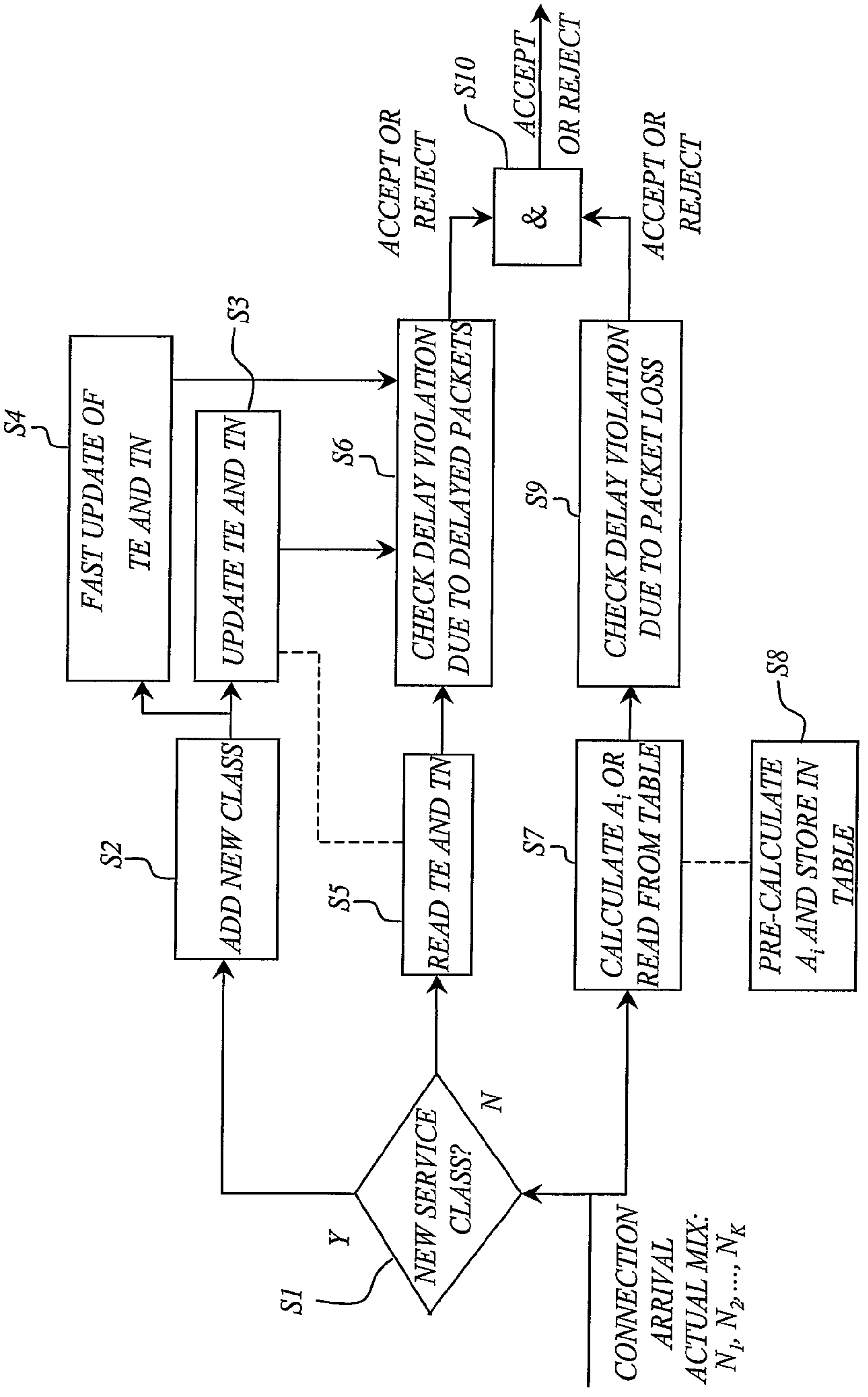

S4
FAST UPDATE OF TE AND TN
S3
UPDATE TE AND TN
S2
ADD NEW CLASS
S1
NEW SERVICE CLASS?
Y
N
CONNECTION ARRIVAL ACTUAL MIX: $N_1, N_2 ..., N_K$
S5
READ TE AND TN
S6
CHECK DELAY VIOLATION DUE TO DELAYED PACKETS
ACCEPT OR REJECT
S10
&
ACCEPT OR REJECT
S9
CHECK DELAY VIOLATION DUE TO PACKET LOSS
ACCEPT OR REJECT
S7
CALCULATE A_i OR READ FROM TABLE
S8
PRE-CALCULATE A_i AND STORE IN TABLE

Fig. 8

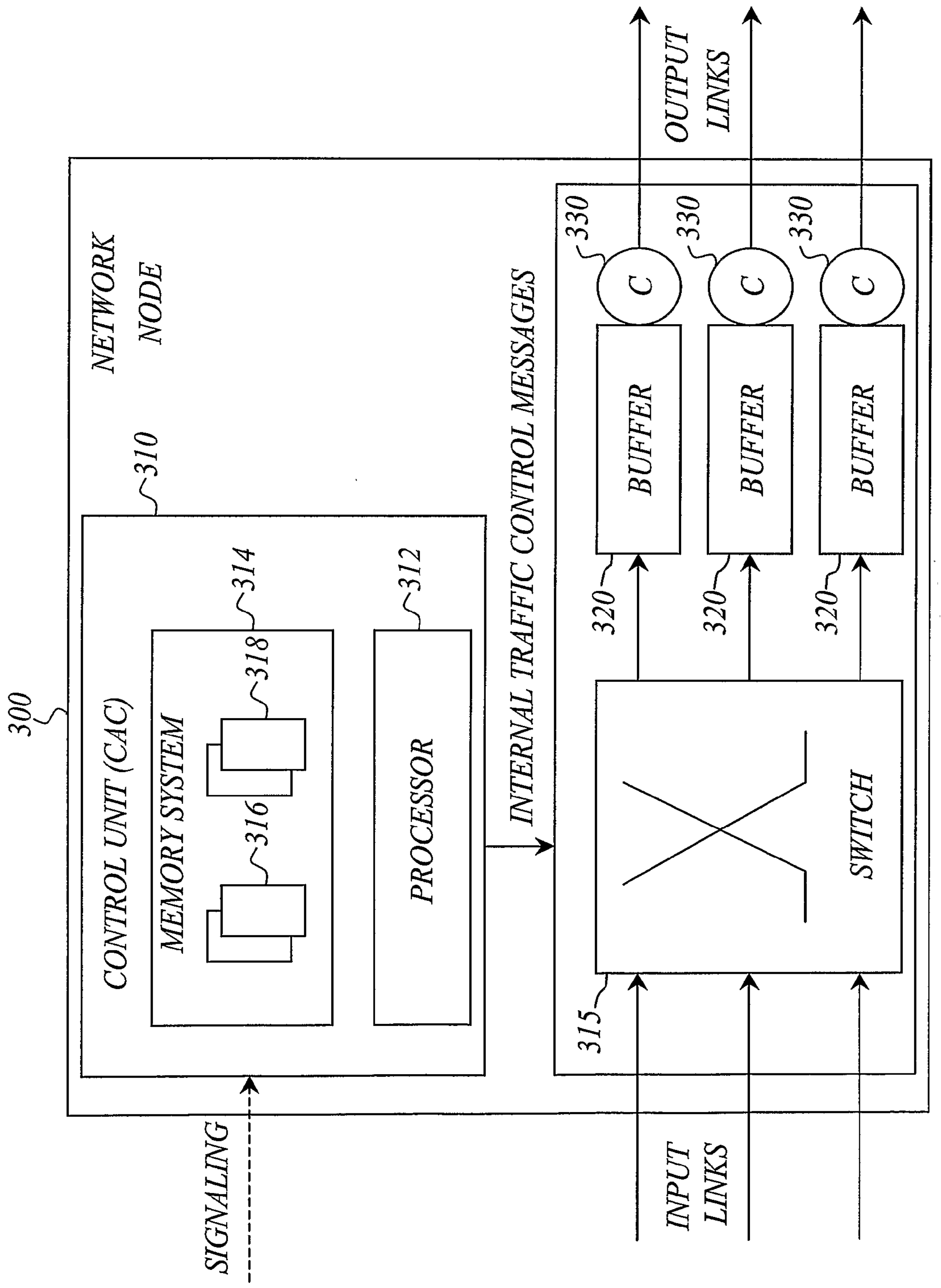

300
NETWORK NODE
310
CONTROL UNIT (CAC)
314
MEMORY SYSTEM
316
318
312
PROCESSOR
SIGNALING
INTERNAL TRAFFIC CONTROL MESSAGES
315
SWITCH
320
320
320
BUFFER
BUFFER
BUFFER
330
330
330
C
C
C
INPUT LINKS
OUTPUT LINKS

Fig. 9

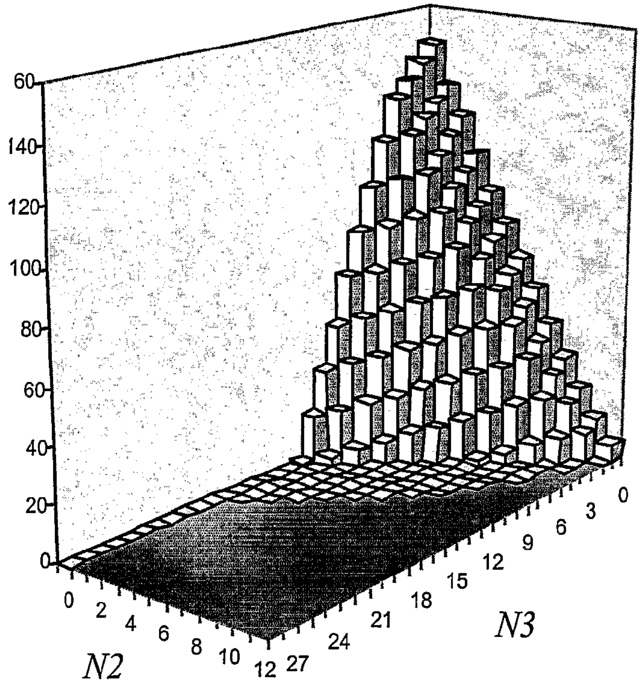


Fig. 10A

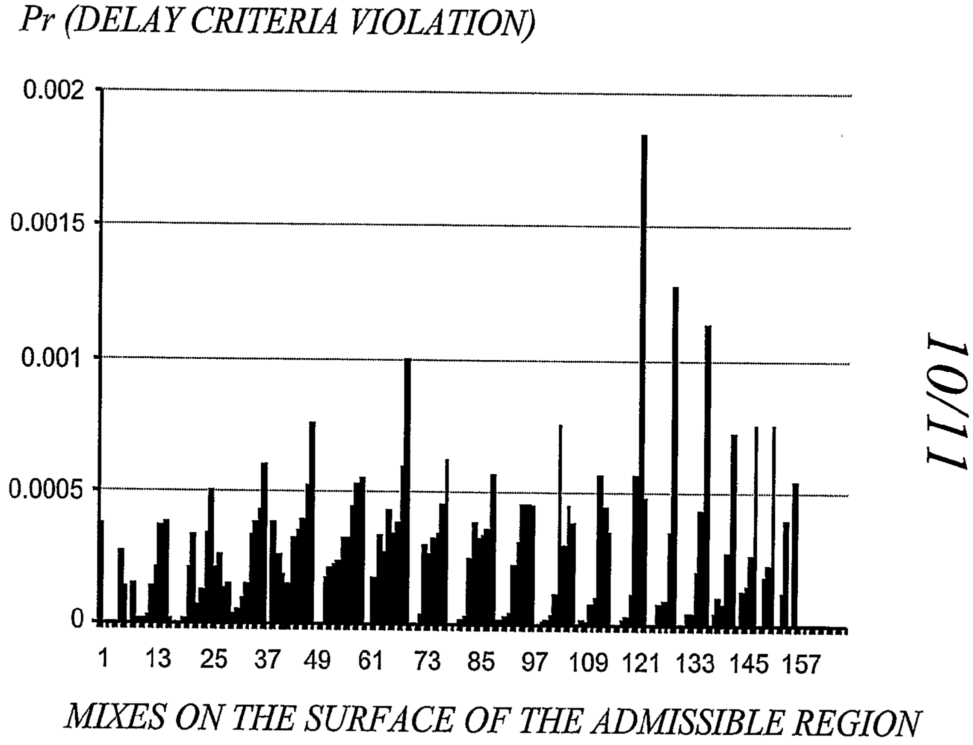


Fig. 10B

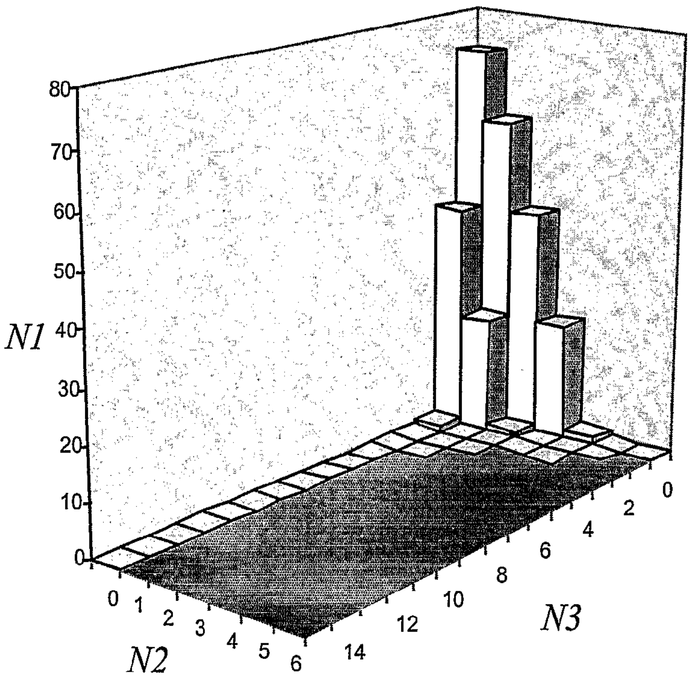


Fig. 11A

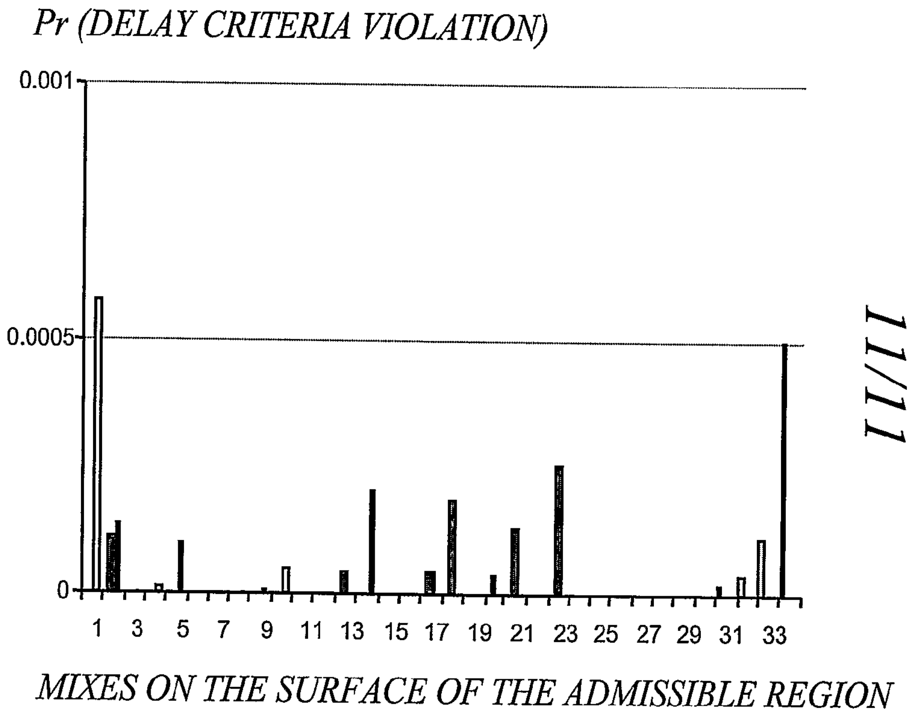


Fig. 11B

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 02/00468

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04Q 7/38, H04L 12/56

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04Q, H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6240066 B1 (NAGARAJAN ET AL), 29 May 2001 (29.05.01), column 6, line 62 - column 7, line 8, figure 3 --	1-6,12-15, 24-31,37-40, 49-50
X	IEEE Journal on Selected Areas in Communications, Vol 11, No 4, May 1993, Jay M Hyman et al: "A Separation Principle Between Scheduling and Admission Control for Broadband Switching", pages 607-611, fig 2 --	1-6,12-15, 24-31,37-40, 49-50
A	IEEE VTS 50th Vehicular Technology Conference,1999 Hyun-Taek Oh et al; "A New Call Admission Control Algorithm for Multi-class Services in Wireless ATM" --	1-50

☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 6 November 2002	Date of mailing of the international search report 0 7 -11- 2002
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Ann Börjeson/EK Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 02/00468

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 0038380 A2 (TELEFONAKTIEBOLAGET LM ERICSSON), 29 June 2000 (29.06.00), page 2, line 12 - page 4, line 4 --	1-50
A	WO 0079829 A1 (NOKIA NETWORKS OY), 28 December 2000 (28.12.00), page 14, line 26 - page 15, line 26 --	1-50
P,A	EP 1217793 A2 (MATSUSHITA ELECTRIC INDUSTRIAL CO, LTD), 26 June 2002 (26.06.02), abstract -- --------	1-50

INTERNATIONAL SEARCH REPORT

Information on patent family members

30/09/02

International application No.

PCT/SE 02/00468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6240066 B1	29/05/01	NONE	
WO 0038380 A2	29/06/00	AU 3093900 A	12/07/00
		CN 1335009 T	06/02/02
		EP 1142225 A	10/10/01
WO 0079829 A1	28/12/00	AU 4613499 A	09/01/01
EP 1217793 A2	26/06/02	JP 2002232467 A	16/08/02
		US 2002120767 A	29/08/02